

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
特開2000-332995  
(P2000-332995A)

(43) 公開日 平成12年11月30日 (2000. 11. 30)

(51) Int.Cl. <sup>7</sup>	識別記号	F I	テーマコード* (参考)
H 0 4 N 1/387	1 0 1	H 0 4 N 1/387	1 0 1 5 B 0 5 7
G 0 6 T 5/00		G 0 6 F 15/68	3 1 0 J 5 C 0 7 6
H 0 4 N 1/407		H 0 4 N 1/40	1 0 1 E 5 C 0 7 7

審査請求 未請求 請求項の数13 O L (全 23 頁)

(21) 出願番号 特願平11-146593

(22) 出願日 平成11年5月26日 (1999. 5. 26)

(31) 優先権主張番号 特願平11-73321

(32) 優先日 平成11年3月18日 (1999. 3. 18)

(33) 優先権主張国 日本 (J P)

(71) 出願人 000006747

株式会社リコー

東京都大田区中馬込1丁目3番6号

(72) 発明者 作山 宏幸

東京都大田区中馬込1丁目3番6号 株式会社リコー内

(74) 代理人 100073760

弁理士 鈴木 誠 (外1名)

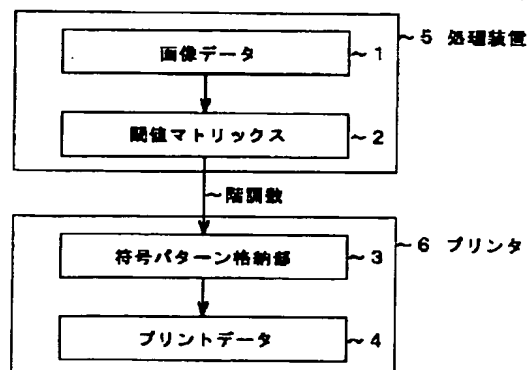
最終頁に続く

(54) 【発明の名称】 画像データ生成方法および記録媒体

(57) 【要約】

【課題】 ラスター展開とイメージの解像度変換が高速で、中間調処理、転送データ量、画質が中程度である、中間調処理と解像度変換を同時に行う。

【解決手段】 閾値マトリックス2の各要素には、複数の閾値が値順にソートされて配置されている。画像データ1の各画素と、画素に対応したマトリックスの要素内の複数の閾値とを比較して、画素値を階調数に変換する。画素と階調数に対応した符号パターン3を選択し、これにより、解像度の高いプリントデータ4を出力する。



## 【特許請求の範囲】

【請求項 1】 第 1 の解像度の画像データから第 1 の解像度より高い第 2 の解像度の画像データを生成する画像データ生成方法であって、前記第 1 の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾値を用いて、画素値を階調数に変換し、前記画素の位置および前記変換された階調数に対応した所定の符号パターンを参照することにより、前記第 2 の画像データを生成することを特徴とする画像データ生成方法。

【請求項 2】 前記階調数変換時に誤差拡散処理を適用することを特徴とする請求項 1 記載の画像データ生成方法。

【請求項 3】 同一階調値における前記符号パターンを、前記第 1 の画像データの画素位置に対応して配置したとき、該配置がドット分散型となることを特徴とする請求項 1 記載の画像データ生成方法。

【請求項 4】 同一階調値における前記符号パターンを、前記第 1 の画像データの画素位置に対応して配置したとき、該配置がブルーノイズ型となることを特徴とする請求項 1 記載の画像データ生成方法。

【請求項 5】 前記階調数変換をホスト側で行い、前記階調数を画像出力装置に対して送信し、前記階調数の送信が  $n$  進法で行われるとき、前記階調数を  $n$  のべき乗とすることを特徴とする請求項 1 記載の画像データ生成方法。

【請求項 6】 第 1 の解像度の画像データから第 1 の解像度より高い第 2 の解像度の画像データを生成する画像データ生成方法において、前記第 1 の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾値を用いて、画素値を階調数に変換し、前記画素の位置および前記変換された階調数に対応した所定の第 1 の符号パターンを参照することにより、前記第 2 の画像データを生成する画像データ生成方法であって、前記注目画素に対応した符号パターンおよび所定個数の前記第 1 の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し一致したとき、前記注目符号パターン群の所定の第 2 の符号パターンを、第 3 の符号パターンに変更することを特徴とする画像データ生成方法。

【請求項 7】 前記階調数変換をホスト側で行い、前記階調数を画像出力装置に対して送信し、該画像出力装置は前記階調数に対応した符号パターンを参照することにより前記第 2 の画像データを生成すると共に、前記第 3 の符号パターンへの変更を行うことを特徴とする請求項 6 記載の画像データ生成方法。

【請求項 8】 前記変更の対象となる前記第 3 の符号パターンは、前記所定の第 1 の符号パターンの何れとも異なることを特徴とする請求項 6 記載の画像データ生成方法。

【請求項 9】 前記注目符号パターン群を複数用いて、

該複数の注目符号パターン群と特定符号パターン群とを比較することを特徴とする請求項 6 記載の画像データ生成方法。

【請求項 10】 前記第 1 の画像データが複数の色成分を有し、該色成分毎に、前記所定形状の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも 1 つの色成分については、前記特定符号パターン群の数を、他の色成分の特定符号パターン群の数よりも少なくしたことを特徴とする請求項 6 記載の画像データ生成方法。

【請求項 11】 前記第 1 の画像データが複数の色成分を有し、該色成分毎に、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも 1 つの色成分については、前記注目符号パターン群の数を、他の色成分の注目符号パターン群の数よりも少なくしたことを特徴とする請求項 9 記載の画像データ生成方法。

【請求項 12】 前記少なくとも 1 つの色成分は黄色であることを特徴とする請求項 10 または 11 記載の画像データ生成方法。

【請求項 13】 第 1 の解像度の画像データから第 1 の解像度より高い第 2 の解像度の画像データを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記第 1 の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾値を用いて、画素値を階調数に変換する機能と、前記画素の位置および前記変換された階調数に対応した所定の符号パターンを参照することにより、前記第 2 の画像データを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】本発明は、画像情報を、出力装置用のデータとして中間調処理および解像度変換する画像データ生成方法および画像データ生成処理プログラムを記録した記録媒体に関し、アプリケーションプログラム、ディスプレイドライバ、プリンタドライバ等のデバイスドライバに適用される技術である。

## 【0002】

【従来の技術】コンピュータの普及により、テキスト、グラフィックス等のベクトルデータや、自然画等のイメージデータを含んだ複雑な文書が容易に作成できるようになった。このような文書を高品位でプリントできるように、プリンタの解像度も向上しているが、その結果、プリンタへ送信すべきデータ量が増大し、処理時間や転送時間の増大を招いている。

【0003】上記したような文書を所定の解像度および階調表現能力を有するプリンタを用いて出力する場合の

代表的な方法（以下、第1の方法）として、例えば特開平5-96792号公報に記載されているものがある。

【0004】この第1の方法では、プリンタドライバ内で、ベクトルデータを所定解像度のビットマップデータとしてラスター展開するとともに、自然画等のイメージデータに対して所定解像度への解像度変換を行った後、両者に対して所定階調数への階調数変換（中間調処理）を施し、階調変換後（中間調処理後）のデータをプリンタに対して送信するものである。従って、プリントスピードを向上させるためには、ラスター展開、解像度変換、および中間調処理の高速化が必要であり、またこれと同時に、プリンタへのデータ転送の高速化も必要である。

【0005】

【発明が解決しようとする課題】そこで、ラスター展開および中間調処理を高速化する方法が、「PPA Printer Software Driver Design」（HP Journal, June 1997）に記載されている。

【0006】この方法（以下、第2の方法）では、ラスター展開および中間調処理を高速化するため、ラスター展開およびイメージデータの解像度変換をプリンタの解像度よりも低い解像度で行い、中間調処理を施した後プリンタに送信し、プリンタのファームウェアを使って全体の解像度変換（イメージデータ単体の解像度変換と区別するため、これを以下“全体の”と呼ぶ）を行って、上記低解像度をプリンタの解像度に変換して出力する（なお、前掲した文献には、テキストをプリンタの1/2の解像度でラスター展開し、ファームウェアを使ってテキストの解像度を倍に変換することが記載されている）。

【0007】上記したように、ラスター展開、ビットマップの解像度変換および中間調処理を低解像度で行えば、処理すべきデータ量が削減される結果、高速化が図られる。しかし、中間調処理後のデータに解像度変換を施した場合、解像度変換を行ってから中間調処理を施す場合に比べて、明らかに画質が低下しやすい。

【0008】以上で説明した第1の方法、第2の方法を簡単に比較すると、以下のような傾向となり、概略「第1の方法は画質優先で、第2の方法は速度優先」と言える。

【0009】

ラスター展開 ; 第1の方法=遅い  
第2の方法=速い  
イメージの解像度変換 ; 第1の方法=遅い  
第2の方法=速い  
中間調処理 ; 第1の方法=遅い  
第2の方法=速い

転送データ量 ; 第1の方法=多い  
第2の方法=少ない  
画質 ; 第1の方法=良い  
第2の方法=悪い。

【0010】本発明は上記した事情を考慮してなされたもので、本発明の目的は、ラスター展開とイメージの解像度変換が従来の第1、第2の方法よりも速く、中間調処理、転送データ量、画質については従来の第1、第2の方法の中間となる、両者の長所を併せ持ち、中間調処理と全体の解像度変換を同時に行う画像データ生成方法および記録媒体を提供することにある。

【0011】すなわち、  
ラスター展開 ; 第1の方法=遅い  
本発明=速い  
第2の方法=速い  
イメージの解像度変換 ; 第1の方法=遅い  
本発明=速い  
第2の方法=速い  
中間調処理 ; 第1の方法=遅い  
本発明=第1と第2の中間  
第2の方法=速い  
転送データ量 ; 第1の方法=多い  
本発明=第1と第2の中間  
第2の方法=少ない  
画質 ; 第1の方法=良い  
本発明=第1と第2の中間  
第2の方法=悪い。

【0012】

【課題を解決するための手段】前記目的を達成するために、請求項1記載の発明では、第1の解像度の画像データから第1の解像度より高い第2の解像度の画像データを生成する画像データ生成方法であって、前記第1の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾値を用いて、画素値を階調数に変換し、前記画素の位置および前記変換された階調数に対応した所定の符号パターンを参照することにより、前記第2の画像データを生成することを特徴としている。

【0013】請求項2記載の発明では、前記階調数変換時に誤差拡散処理を適用することを特徴としている。

【0014】請求項3記載の発明では、同一階調値における前記符号パターンを、前記第1の画像データの画素位置に対応して配置したとき、該配置がドット分散型となることを特徴としている。

【0015】請求項4記載の発明では、同一階調値における前記符号パターンを、前記第1の画像データの画素位置に対応して配置したとき、該配置がブルーノイズ型となることを特徴としている。

【0016】請求項5記載の発明では、前記階調数変換をホスト側で行い、前記階調数を画像出力装置に対して送信し、前記階調数の送信がn進法で行われるとき、前

記階調数を $n$ のべき乗とすることを特徴としている。

【0017】請求項6記載の発明では、第1の解像度の画像データから第1の解像度より高い第2の解像度の画像データを生成する画像データ生成方法において、前記第1の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾値を用いて、画素値を階調数に変換し、前記画素の位置および前記変換された階調数に対応した所定の第1の符号パターンを参照することにより、前記第2の画像データを生成する画像データ生成方法であって、前記注目画素に対応した符号パターンおよび所定個数の前記第1の画像データに対応した符号パターンから構成される所定形状の注目符号パターン群と、特定符号パターン群とを比較し一致したとき、前記注目符号パターン群の所定の第2の符号パターンを、第3の符号パターンに変更することを特徴としている。

【0018】請求項7記載の発明では、前記階調数変換をホスト側で行い、前記階調数を画像出力装置に対して送信し、該画像出力装置は前記階調数に対応した符号パターンを参照することにより前記第2の画像データを生成すると共に、前記第3の符号パターンへの変更を行うことを特徴としている。

【0019】請求項8記載の発明では、前記変更の対象となる前記第3の符号パターンは、前記所定の第1の符号パターンの何れとも異なることを特徴としている。

【0020】請求項9記載の発明では、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較することを特徴としている。

【0021】請求項10記載の発明では、前記第1の画像データが複数の色成分を有し、該色成分毎に、前記所定形状の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも1つの色成分については、前記特定符号パターン群の数を、他の色成分の特定符号パターン群の数よりも少なくしたことを特徴としている。

【0022】請求項11記載の発明では、前記第1の画像データが複数の色成分を有し、該色成分毎に、前記注目符号パターン群を複数用いて、該複数の注目符号パターン群と特定符号パターン群とを比較するとき、前記色成分の内、少なくとも1つの色成分については、前記注目符号パターン群の数を、他の色成分の注目符号パターン群の数よりも少なくしたことを特徴としている。

【0023】請求項12記載の発明では、前記少なくとも1つの色成分は黄色であることを特徴としている。

【0024】請求項13記載の発明では、第1の解像度の画像データから第1の解像度より高い第2の解像度の画像データを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体であって、前記第1の画像データの注目画素毎に、該画素の位置に対応した、ソートされた複数の閾

値を用いて、画素値を階調数に変換する機能と、前記画素の位置および前記変換された階調数に対応した所定の符号パターンを参照することにより、前記第2の画像データを生成する機能をコンピュータに実現させるためのプログラムを記録したコンピュータ読み取り可能な記録媒体であることを特徴としている。

【0025】

【発明の実施の形態】以下、本発明の一実施例を図面を用いて具体的に説明する。

（実施例1）図1は、本発明の実施例1の構成を示す。図において、1はラスター展開または解像度変換した後の画像データ、2は閾値が配列された閾値マトリックス、3は符号パターンが格納された符号パターン格納部、4はプリントデータであり、画像データ1と閾値マトリックスは処理装置5内にあり、符号パターン格納部3とプリントデータ4はプリンタ6内にある。また、本発明の実施例1では、例えば解像度600dpi、階調数2値のプリンタを出力装置として用いる。

【0026】以下、本発明を説明する前に前述した従来の第1、第2の方法について図を用いて説明する。図2は、階調数変換（中間調処理）で使用する4×4の閾値マトリックスを示す。マトリックスのデータの各画素は、例えば0から15までの値（各画素が4ビットの多値）をとる。

【0027】図3（a）は、前記した第1の方法によって、ベクトルデータを600dpiでラスター展開した後のデータ、あるいはイメージデータを600dpiに解像度変換した後のデータである。また、図4（a）は、同様に前記した第2の方法によって、300dpiでラスター展開または解像度変換した後のデータを示す。なお、この例では、図3（a）は、図4（a）を公知の最近傍法で2倍に解像度変換した場合と同じものになっている。

【0028】ラスター展開（あるいは解像度変換）自体の処理量は、概ね解像度の2乗に比例するので、第1の方法のラスター展開（あるいは解像度変換）の処理量は、第2の方法の処理量の4倍となる。

【0029】第1の方法においては、図3（a）のデータについて各画素値と、各画素位置に対応する図3

（b）（図2と同一のマトリックス）の閾値マトリックスの閾値とを比較し、公知のディザ法によって2値化を行う。この例の場合、閾値以上の値を有する画素の値を1と出力するので、階調値として図3（c）を得るが、閾値に対して計16回の比較演算が必要となる。該データに対して、圧縮等を行わない場合には、図3（d）に示すデータがプリントすべきデータ（図3（c）と同一）となるため、プリンタに対しては、16画素×2bit=32bitのデータが転送されることになる。

【0030】同様に、第2の方法では、図4（a）のデータに対して、図2の閾値マトリックスを作用させる場

合は、 $4 \times 4$ のマトリックスの $1/4$ のマトリックスが順に使用され、この例では図4(b)のマトリックスによって2値化を行って、図4(c)の階調値を得る(図4(a)のデータの右に隣接する $2 \times 2$ のデータ(図は省略)に対しては、図2中で図4(b)のマトリックスの右に隣接する $2 \times 2$ のマトリックス(閾値3、15、11、7)が使用される)。

【0031】この第2の方法の場合、閾値に対して計4回の比較演算しか行われないので、第1の方法に比べて中間調処理に要する時間は少ない。また、データ圧縮等を行わない場合には、プリンタに対して4画素 $\times$ 2bit = 8bitのデータが転送されることになり、転送データ量も少ない。

【0032】ただし、第2の方法においては、プリンタ側のファームウェアで、図4(c)のデータ全体の解像度を2倍にするため、プリントすべきデータは図4(d)となる。なお、この例では、解像度変換法として公知の最近傍法を用いる。

【0033】図5は、本実施例を説明する図であり、図5(a)は本実施例の画像データ1であり、前述した第2の方法と同様に、300dpiでラスター展開または解像度変換を行った後のデータ( $2 \times 2$ 画素)である。また、本実施例の閾値マトリックス2は、図5(b)のように閾値が配列されている。そして、画像データ1を、閾値マトリックス2によって0から4までの階調数に変換する。

【0034】図5(b)は、要素が4個の配列の $2 \times 2$ のマトリックスであり、配列を構成する4つの閾値は、図2のマトリックスを $2 \times 2$ のサブマトリックス4つに分解したときの各サブマトリックス内の閾値と同じであるが、閾値が値順にソートされている。

【0035】本実施例における階調数変換は、図5(a)の1画素のデータと、その画素位置に対応する4個の閾値とを比較して(つまり、図5(b)のソートされた閾値を順に比較して)、閾値が画素データを越えるまでの(比較回数-1)を階調値とする。すなわち、例えば、画像データ(15)と、その対応する閾値マトリックスの閾値(1、5、9、13)とを比較し、画像データ(7)と、それに対応する閾値マトリックスの閾値(3、7、11、15)とを比較する。

【0036】この結果、図5(a)のデータに対応する階調数として図5(c)が得られ、その比較回数は計11回となる。

【0037】また、本実施例においては、図5(c)に示した階調値をプリンタ6に対して送信するが、そのデータ量は、4画素 $\times$ 3bit(5階調) = 12bitである。

【0038】そして、本実施例においては、プリンタ6側には、 $2 \times 2$ の符号パターンである5つの配列を要素とし、図5(a)の4つの画素の各位置に対応した $2 \times$

2の符号パターンマトリックス(図6)が符号パターン格納部3に設けられている。

【0039】図6の符号パターンマトリックスの各要素は、図2のマトリックスを $2 \times 2$ のサブマトリックス4つに分解し、各サブマトリックスに $2 \times 2$ のデータ(1画素16階調)を作用させたときに、とり得る $2 \times 2$ のパターンを、階調数順にソートしたものである。符号パターンは、ベクトル量子化におけるコードブックに相当するものである。

【0040】そして、図5(a)の値15の注目画素(\*印)に対しては、図6の中の太線内(\*印)の5つの符号パターンが対応し、ファームウェア中で、注目画素の階調値4に対応した二重線で囲まれた符号パターン(1、1、1、1)が選択される。他の画素も同様に、画素の位置と階調値に対応した符号パターンが選択され、最終的にプリントすべきデータ4として、図5(d)が得られる。

【0041】本実施例では、第2の方法で図4(d)の最終データを得るための全体の解像度変換に相当する処理を、符号パターンの参照によって行うため、“全体の解像度変換”にかかる処理は、第2の方法よりも高速ではない。しかし、図4(d)、3(d)、5(d)を比較すれば明らかなように、本実施例の方法による最終データ図5(d)は、第2の方法よりも高画質(例えば、原データの階調性がよりよく保存されている)で、第1の方法に匹敵するものである(本実施例では、簡単のため、図3と図4を最近傍法で解像度変換した関係にしたが、両者の関係がそうでない場合には、本実施例の最終データは第2の方法よりは劣ることになる)。

【0042】以上の例において、3つの方法の処理量の比較結果を簡単にまとめると、以下ようになる(中間調処理の処理量は、閾値との比較回数で近似している)。

【0043】

ラスター展開	；第1の方法=4
	本発明=1
	第2の方法=1
イメージの解像度変換	；第1の方法=4
	本発明=1
	第2の方法=1
中間調処理	；第1の方法=16
	本発明=11
	第2の方法=4
転送データ量	；第1の方法=32
	本発明=12
	第2の方法=8
全体の解像度変換	；第1の方法=不要
	本発明=中速
	第2の方法=高速

画質 ; 第1の方法=良い  
 本発明=中〜良  
 第2の方法=悪い。

従って、本実施例は、第1の方法と第2の方法の両方の長所を併せ持った方法であるといえることができる。

【0044】図7は、本発明の実施例1の処理フローチャートである。x行×y列の画像データを、 $k^2$  ( $k$ の2乗) 個のソートされた閾値の配列を要素とするm行×n列の閾値マトリックスで階調数変換して階調数を得て(ステップ102)、 $k \times k$ の符号パターン $k^2 + 1$ 個の配列を要素とするm行×n列の符号パターンマトリックスを参照して、 $k \times$ 行× $k \times$ 列のプリントデータを得る(ステップ103)。マトリックスの右端、下端に達したときは(ステップ106、110)、それぞれ画像データを $n+1$ 画素、 $m+1$ 画素ずらした位置から同様に処理し、画像データの下端に至るまで(ステップ109)同じ処理を繰り返す。

【0045】図中、 $D(i, j)$ は、画像データ中の(行、列) =  $(i, j)$ に位置するデータであり、 $M(p, q)$ は、閾値マトリックス中の(行、列) =  $(p, q)$ に位置する閾値配列であり、 $t(i, i)$ は、 $D(i, i)$ に $M(p, q)$ を作用させたときに得られる階調数であり、 $P(p, q, t(i, j))$ は、 $D(i, j)$ に対して参照すべき、符号パターンマトリックス中の1つの符号パターンである。

【0046】また、ソートされた閾値の配列を要素とするm行×n列の閾値マトリックスは、 $k$ m行× $k$ n列の閾値マトリックスから図8の処理フローによって作成でき、m行×n列の符号パターンマトリックスは、 $k$ m行× $k$ n列の閾値マトリックスから、図9の処理フローによって作成できる。

【0047】なお、実施例1においては、原データの縦横比を維持するため、符号パターンの大きさを $k \times k$ としているが、維持の必要がない場合には、縦横比が1でない符号パターンを作成すればよい。

【0048】(実施例2) 実施例1では、 $k^2$ 個のソートされた閾値の配列を用いて $k^2 + 1$ 階調への変換を行った。通常、プリンタへのデータ転送は2進法で行われるため、ここで階調数が2のべき乗になるように構成すると、(転送データ/階調数)が最小となり転送効率がい

【0049】図10は、実施例2を構成する閾値マトリックスであり、3個のソートされた閾値の配列を用いて4階調(0から3)への変換を行えば、4画素分の階調データは8bit (= 4画素×2bit (4階調))で済むため、(転送データ/階調数)を実施例1よりも小さくすることができる。なお、図10の各サブマトリックスは、図5(b)の各サブマトリックスの閾値配列の中央2つの値を、該2つの値の平均値1つで置き換えたものである。

【0050】(実施例3) 上記した実施例は、ディザ法による中間調処理を第2の方法よりも高速に行うものであるが、公知の誤差拡散法にも適用可能である。図11は、本発明を誤差拡散法に適用した場合の実施例3を説明する図である。図11(a)は、実施例3で使用する周知の誤差マトリックスであり、\*印の注目画素(図5(a)の画素値15)の階調数変換で生じた誤差( $15 - 13 = 2$ )を、図11(b)のように注目画素に隣接する右および下の画素に半分ずつ加算し、図11(c)に示すデータの注目画素(\*印)に対して、さらに階調数変換を行う。本実施例でも、図6と同様な符号パターンマトリックスが使用できる。図12は、実施例3の処理フローを示し、図7の処理にさらに誤差を配分するステップ204が追加されている。

【0051】(実施例4) 実施例1は、図2のマトリックスを基に符号パターンマトリックスを生成したが、図13に示すような典型的なドット分散型の閾値マトリックスを基に符号パターンマトリックスを生成すると、図14のように、同一階調値における符号パターンの配列自体もドット分散型となる。すなわち、図14において、例えば階調値1に対応した4つの符号パターンを、原画像データの画素位置に対応させて配置すると、 $4 \times 4$ のマトリックスができるが、これはドット分散型である。なお、図2のマトリックスもドット分散型的な性質を有する。

【0052】図14から明らかなように、符号パターンマトリックスを構成する符号パターンの配列は全て同じとなるから、図14の例の場合、プリンタ側で5種類の符号パターンのみを保持することにより、符号パターンマトリックスの容量を小さくできると同時に、マトリックスの参照も高速となる。

【0053】(実施例5) また、本発明では、図示しない公知のブルーノイズ型の閾値マトリックス(閾値の配列がランダム)を基に符号パターンマトリックスを生成することも可能である。この場合には、同一階調値における符号パターンの配列自体もブルーノイズ型となる。そして、本実施例では、ディザ法と同様の処理速度で、誤差拡散に匹敵し得る画質を得ることができる。

【0054】なお、ブルーノイズ型の閾値マトリックスに関しては、「Digital Halftoning Using a Blue Noise Mask」, T. Mitsa and K. Parker, 1991 International Conference on Acoustics, Speech, and Signal Processing, February and May 1991 (ブルーノイズマスク法の原理など) および「Modified Approach to the Construction of a Blue Noise Mask」by M. Yao and K. Parker, Journal of

Electronic Imaging, Vol. 3, No. 1, 92-97, January 1994 (ブルーノイズマスクの作成法等)を参照されたい。

【0055】(実施例6)上記した実施例では、プリンタ6側が符号パターンマトリックス3を保持していたが、処理装置5(ホスト)側から符号パターンマトリックス3をダウンロードするように構成してもよい。また、階調数変換に使用する閾値マトリックスを切り替える場合には、符号パターンの変更を伴う。これらの場合ホストが、階調データの送信に先立ち、符号パターンマトリックスをプリンタに送信すればよい。

【0056】図15は、実施例6の処理フローチャートである。プリントを開始する前に、ホスト5は符号パターンが送信済みでなく(ステップ301)、符号パターンに変更があるとき(ステップ302)、符号パターンをプリンタ6に送信し(ステップ303)、その後、ホストがプリンタに階調値を送信し(ステップ304)、プリンタ6は送信された階調値に対応した符号パターンを参照してプリントデータを出力する(ステップ305)。

【0057】このように、本実施例では、階調数をプリンタなどの画像出力装置に対して送信するので、転送データ量が削減でき、また、符号パターンをダウンロード可能にしたので、符号パターンの変更に対応することができる。

【0058】(実施例7)ところで、前述した第2の方法は、例えば特開平2-112966号公報に示されるようなビットパターンのテンプレートマッチングを行うことによって、ジャギーのスムージングを行うことが可能である。しかし、ビットパターン同士でテンプレートマッチングを行う場合には、処理量が多くなり、速度低下を招きやすいという問題がある。

【0059】そこで、本実施例では、後述する符号パターンのテンプレートマッチングを行い、簡易な処理を行うことによって、高速にジャギーのスムージングを行う方法を提案する。

【0060】以下、本実施例について説明するが、本実施例においても、実施例1と同様に出力装置として解像度600dpi、階調数2値のプリンタを用いる。また、想定する閾値マトリックスは図2と同様であり、図14と同じ符号パターンマトリックスを使用する。ただし、本実施例では、図16に示すように、異なる符号パターンにパターン番号(0、1、2、3a、3b、4)を付与したものをを用いる。また、原画像データとしては、黒色のみのモノクロ画像を想定する。

【0061】さて、図17、18は、「文」という黒文字のベクトルフォントを300dpiでラスター展開し、実施例1の処理手順によって得られたビットパターン(符号パターン)を示す(なお、本来は1枚の図であるが、これを2つに分割し、図17は「文」の左半分で

あり、図18は「文」の右半分である)。グレーの部分には符号パターン中の値1が、白い部分には符号パターン中の値0が対応していて、この例では濃い黒色の文字の処理結果であることから、階調値としては0または4のいずれかとなっている。

【0062】図17の左上(a)には、300dpi、600dpiの各画素のサイズが示されていて、(b)には、600dpiの画素サイズに対応した2×2の符号パターンの並び、およびパターン番号の並びが示されている。また、(b)中には、本実施例における、符号パターンのサイズおよび符号パターン群のサイズが示されている。

【0063】この例においては、注目符号パターンを含む4つの符号パターンから成る正方形の注目符号パターン群を、6種類の正方形の特定符号パターン群(図19)と照合し、一致した場合には、注目符号パターン群をそれぞれ図20に示す符号パターン群に変更する。図20において黒色で示した部分は、符号パターン中の変更された位置を示すが、図16から明らかなように、図20の変更後の符号パターン(黒色を含む符号パターン)の内の3つは、図16には存在しない。つまり、図20(e)の符号パターンは、図16のパターン番号1として存在するが、図20(c)、(d)、(f)の符号パターンは、図16のパターン中にはない。

【0064】従って、本実施例では、図21に示す3つの符号パターン(パターン番号1b、1c、1d)を追加して使用する。また、図16から明らかなように、符号パターンには重複が生じるので、予め重複を排除したパターン番号を図16、図21に示すように定義する。なお、符号パターン数の増加が好ましくない場合には、符号パターンを追加しないという選択も可能である。

【0065】ここで、図19(a)の注目符号パターン群を例にとると、図19と図20の関係は以下の通りである。図19(a)の特定符号パターン群は、第1象限から左回りに0、4、0、4というパターン番号の順列を形成する。そこで、注目符号パターン群が、第1象限から左回りに0、4、0、4というパターン番号列を形成した場合には、これを図20(a)に示す1d、4、1b、4というパターン番号列(符号パターン群)に変更する。

【0066】また、この例においては、1つの注目符号パターン(図17(c)の\*印)に対する注目符号パターン群(比較のためのウインドウ)のとり方として、図17(c)に示すA、B、C、Dの4種類(太線A、二重線B、点線C、三重線D)を採用し、注目符号パターン自体のとり方としては、図17(d)の\*印に示す通り、縦方向・横方向ともに1つ置きとしている(1つ置きで十分だからである)。図17、18全体に配置された格子は、図17(d)の\*印に対する、注目符号パターン群Aのとりかたと一致している。

【0067】本実施例においては、1つの注目符号パターンに対し、注目符号パターン群（比較のためのウィンドウ）Aをとり、特定符号パターン群（図19）と一致した場合には注目符号パターン群を図20に変更して、さらに注目符号パターン群Bと比較し、といった処理を繰り返す。

【0068】なお、本実施例においては、注目符号パターン群Aにおける符号パターンの変更が、注目符号パターン群Bにおける比較・変更に影響を与えないため、1つの注目符号パターンに対してA、B、C、Dととって比較・変更した後、次の注目符号パターンに移るという処理方法でも、全ての注目符号パターンに対してAをとって比較し、全ての注目符号パターンに対してBをとって比較し、という処理方法でも結果は同じである。

【0069】そこで、図22は、全ての注目符号パターンについて、注目符号パターン群（比較のためのウィンドウ）Aをとった場合の処理結果を示したものであり、黒い部分が変更された符号パターンを示す。これによって、ジャギーのスムージングがなされ、比較のためのウィンドウが1つだけの場合でもある程度の効果があることが分かる。

【0070】同様に、図23は、全ての注目符号パターンについて、注目符号パターン群（比較のためのウィンドウ）Bをとった場合の処理結果を示したものであり、図17、18全体に配置された格子は、図17（d）の＊印に対する、注目符号パターン群Bのとりかたと一致している。図中の11から25部分が新たに変更された符号パターンを示す。

【0071】同様に、注目符号パターン群（比較のためのウィンドウ）Cをとった処理結果が図24、注目符号パターン群（比較のためのウィンドウ）Dをとった処理結果が図25図であり、それぞれ31から48の部分、51から62の部分が新たに変更された符号パターンを示す。比較のためのウィンドウを複数使用する分、スムージングの効果も大きくなっているのが明らかである。

【0072】このように本実施例によれば、簡単かつわずか6つの特定パターンの比較・検出という簡易な方法によって、かなりのスムージング効果が発揮されることがわかる（もちろん、スムージングによって弊害が生じる部分も存在するが、全体の効果からみればわずかなものである）。そして、簡易な方法であるので、高速にスムージングを行うことが可能になる。

【0073】なお、原画像データの端部においては、注目符号パターン群（比較のためのウィンドウ）を4つ全てとることができない場合があるが、その場合はとり得る注目パターン群のみ比較すればよい。

【0074】図26は、以上で説明した実施例7の処理フローチャートである。まず、実施例1で説明した方法によって、注目画素に対応する階調数を得る（ステップ401）。上記した階調数変換をホスト側で行い、階調

数をホストからプリンタに送信する（ステップ402）。

【0075】以下の処理はプリンタによって行われる。すなわち、階調数に対応した符号パターンマトリックス（図16）を参照して、注目画素のパターン番号を得る（ステップ403）。上記した処理を全ての注目画素について行う（ステップ404）。

【0076】注目符号パターンについて、A～Dの全ての注目符号パターン群と全ての特定符号パターン群（図19）を比較し（ステップ405）、一致したとき（ステップ406）、注目符号パターン群中の所定符号パターン（パターン番号）を変更する（ステップ407）。

【0077】例えば、注目画素のパターン番号が0（白）で、注目符号パターンが図19の（f）に一致したとき、注目画素のパターン番号0を、図20（f）に変更し、つまりパターン番号1b（図21）に変更する。また、例えば、注目画素のパターン番号が4（黒）で、注目符号パターンが図19の（d）に一致したとき、注目符号パターン中のパターン番号0を、パターン番号1d（図21）に変更する。上記した処理を全ての注目符号パターンについて行う（ステップ408）。

【0078】注目画素のパターン番号（図16）と変更されたパターン番号（図21）から、注目画素に対応する600dpiの最終画像データを得る（ステップ410）。

【0079】（実施例8）上記した実施例では、黒色のみのモノクロ画像を想定したが、カラー画像の場合には公知のように、以上の処理を各画像成分毎に行えばよい。すなわち、300dpiでラスター展開または解像度変換を行った後のデータを、公知の方法で各色成分、例えばC（シアン）、M（マゼンタ）、Y（黄色）、K（黒）毎のプレーンに分離し、各プレーン毎に図26の処理を行えばよい。

【0080】ここで、例えば上記4プレーンの全てに関して、6つの特定符号パターン群全て（図19）を用いた比較を行っても良いが、スムージング効果よりも高速性を求める場合、視認性の低い黄色については、特定符号パターン群の数を減らしてもよい。すなわち、特定符号パターン群のサブセットを使用する。例えば、黄色に関しては、図19（a）、（b）に示した2つの特定符号パターン群との比較のみに止める。この場合の処理フローは、黄色のみに関して、図26における比較ステップ405を「注目符号パターンについて全ての注目符号パターン群と、特定符号パターン群のサブセットを比較（パターン番号列同士を比較）」と置き換えればよい。

【0081】あるいは、黄色に関しては、上記4つの注目符号パターン群A、B、C、D（図17（c））を用いた比較を行わずに、Aのみを用いた比較に止めても良い。前述したように、比較のためのウィンドウが1つだけの場合でもスムージング効果があるからである。この



場合の処理フローは、黄色のみに関して、図26における比較ステップ405を「注目符号パターンについて、注目符号パターン群Aと、全ての特定符号パターン群を比較（パターン番号列同士を比較）」と置き換えればよい。

【0082】上記した例では、視認性の低い黄色について注目符号パターン群の数を少なくしたので、ユーザによって観察されるスムージング効果の低下を抑えながら処理速度を向上させることができる。そして、処理速度を最優先させるならば、黄色については特定符号パターン群との比較を行わないという選択も可能である。

【0083】なお、上記した実施例では文字への適用について示したが、本発明はこれに限らず、グラフィックス画像や自然画等にも適用可能である。

【0084】（実施例9）本発明は上記した実施例に限定されず、ソフトウェアによっても実現することができる。本発明をソフトウェアによって実現する場合には、図27に示すように、CPU、メモリ、表示装置、ハードディスク、キーボード、CD-ROMドライブ、スキャナなどからなるコンピュータシステムを用意し、CD-ROMなどのコンピュータ読み取り可能な記録媒体には、本発明の画像データ生成機能を実現するプログラムなどが記録されている。そして、本発明の画像データ生成処理を実行させるときは、上記した記録媒体に記録されたプログラムを読み出し、これをプリンタドライバとしてOSの一部に組み込むことによって本発明の処理機能が実現される。また、処理対象となる原画像データは例えばハードディスクに格納されていて、これに上記した処理が施され、原画像データよりも解像度の高い画像データがプリンタなどに出力される。

【0085】

【発明の効果】以上、説明したように、請求項1、13記載の発明によれば、原画像データの注目画素に対して、ソートされた閾値を用いて階調数変換を行い、符号パターンを参照するので、原画像データよりも解像度の高い画像データを高速に生成することができ、従来の第2の方法よりも高画質な画像を得ることができる。

【0086】請求項2記載の発明によれば、階調数変換時に誤差拡散処理を適用するので、高画質な画像データを生成することができる。

【0087】請求項3記載の発明によれば、同一階調値における符号パターンのドット配置がドット分散型の配置をとるようにしたので、符号パターンの数を少なくすることができる。

【0088】請求項4記載の発明によれば、同一階調値における符号パターンのドット配置がブルーノイズ型の配置をとるようにしたので、高速かつ誤差拡散処理に匹敵する高画質な画像データを生成することができる。

【0089】請求項5記載の発明によれば、階調数の送信がn進法でなされる場合に、階調数をnのべき乗とし

たので、階調数当たりの転送データ量を最小にすることができる。

【0090】請求項6記載の発明によれば、テンプレートマッチングによって、注目符号パターン群の所定の第2の符号パターンを、第3の符号パターンに変更するので、原画像データよりも解像度が高くかつスムージングのかかった画像データを高速に生成することができる。

【0091】請求項7記載の発明によれば、階調数変換をホストコンピュータ側で行い、階調数を画像出力装置に対して送信し、画像出力装置が符号パターンを参照して原画像データよりも解像度の高い画像データを生成すると共に、第3の符号パターンへの変更を行うので、ホストからプリンタへの転送データ量が削減でき、かつスムージングのかかった最終画像データを高速に生成することができる。

【0092】請求項8記載の発明によれば、変更の対象となる第3の符号パターンが、画素に対応した符号パターンのいずれとも異なるので、符号パターンが拡張される結果、高いスムージング効果を得ることができる。

【0093】請求項9記載の発明によれば、注目符号パターン群を複数用いるので、高いスムージング効果を得ることができる。

【0094】請求項10記載の発明によれば、原画像データの色成分の少なくとも1つについて、特定符号パターン群の数を少なくしたので、スムージング効果への影響を抑えながら処理速度を高めることができる。

【0095】請求項11記載の発明によれば、原画像データの色成分の少なくとも1つについて、注目符号パターン群の数を少なくしたので、スムージング効果への影響を抑えながら、処理速度を高めることができる。

【0096】請求項12記載の発明によれば、1つの色成分を視認性の低い黄色としたので、ユーザによって観察されるスムージング効果の低下を抑えながら、処理速度を向上させることができる。

【図面の簡単な説明】

【図1】本発明の実施例1の構成を示す。

【図2】階調数変換（中間調処理）で使用する4×4の閾値マトリックスを示す。

【図3】（a）～（d）は、従来の第1の方法を説明する図である。

【図4】（a）～（d）は、従来の第2の方法を説明する図である。

【図5】（a）～（d）は、本実施例を説明する図である。

【図6】符号パターンマトリックスを示す。

【図7】本発明の実施例1の処理フローチャートである。

【図8】本発明の実施例1の閾値マトリックス作成の処理フローチャートである。

【図9】本発明の実施例1の符号パターンマトリックス

作成の処理フローチャートである。

【図10】本発明の実施例2を構成する閾値マトリックスである。

【図11】本発明を誤差拡散法に適用した場合の実施例3を説明する図である。

【図12】本発明の実施例3の処理フローチャートである。

【図13】ドット分散型の閾値マトリックスを示す。

【図14】本発明の実施例4の符号パターンマトリックスを示す。

【図15】本発明の実施例6の処理フローチャートである。

【図16】本発明の実施例7で使用する符号パターンマトリックスを示す。

【図17】「文」という黒文字のベクトルフォントをラスタ展開し、実施例1の処理手順によって得られた符号パターンの左半分を示す。

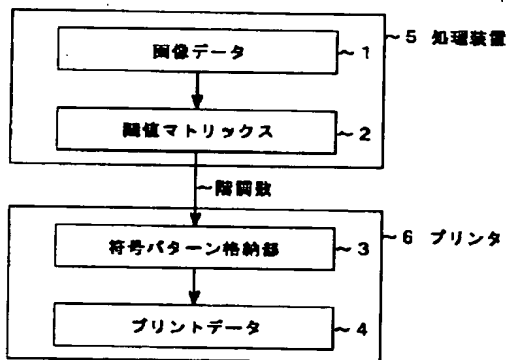
【図18】「文」の右半分を示す。

【図19】(a)～(f)は、特定符号パターン群を示す。

【図20】(a)～(f)は、変更後の符号パターン群を示す。

【図21】追加された符号パターンとパターン番号を示す。

【図1】



【図10】

1, 7, 13	3, 9, 15
4, 10, 16	2, 8, 14

す。

【図22】図17、18の全ての注目符号パターンについて、比較のためのウィンドウAをとった場合の処理結果を示す。

【図23】図17、18の全ての注目符号パターンについて、比較のためのウィンドウBをとった場合の処理結果を示す。

【図24】図17、18の全ての注目符号パターンについて、比較のためのウィンドウCをとった場合の処理結果を示す。

【図25】図17、18の全ての注目符号パターンについて、比較のためのウィンドウDをとった場合の処理結果を示す。

【図26】本発明の実施例7の処理フローチャートである。

【図27】本発明の実施例9の構成を示す。

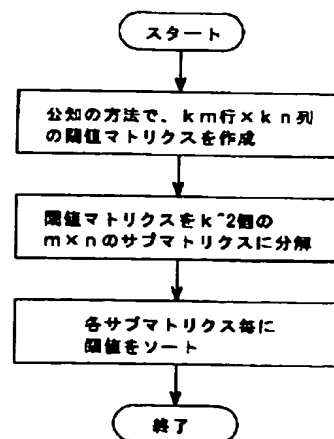
【符号の説明】

- 1 画像データ
- 2 閾値マトリックス
- 3 符号パターン格納部
- 4 プリントデータ
- 5 処理装置
- 6 プリンタ

【図2】

1	9	3	15
13	5	11	7
4	16	2	10
12	8	14	6

【図8】



【図3】

データ				閾値マトリクス			
15	15	7	7	1	9	3	16
15	15	7	7	13	6	11	7
1	1	4	4	4	16	2	10
1	1	4	4	12	8	14	6

(a)

(b)

【図4】

15	7	1	9
1	4	13	5

(a)

(b)

階調値				最終データ			
1	1	1	0	1	1	1	0
1	1	0	1	1	1	0	1
0	0	1	0	0	0	1	0
0	0	0	0	0	0	0	0

(c)

(d)

1	0	1	1	0	0
0	0	0	0	0	0

(c)

(d)

【図5】

データ		閾値マトリクス	
15	7	1, 6, 9, 13	3, 7, 11, 15
*			
1	4	4, 8, 12, 16	2, 6, 10, 14

(a)

(b)

階調値		最終データ	
4	2	1	1
0	1	1	0
		0	0
		0	0

(c)

(d)

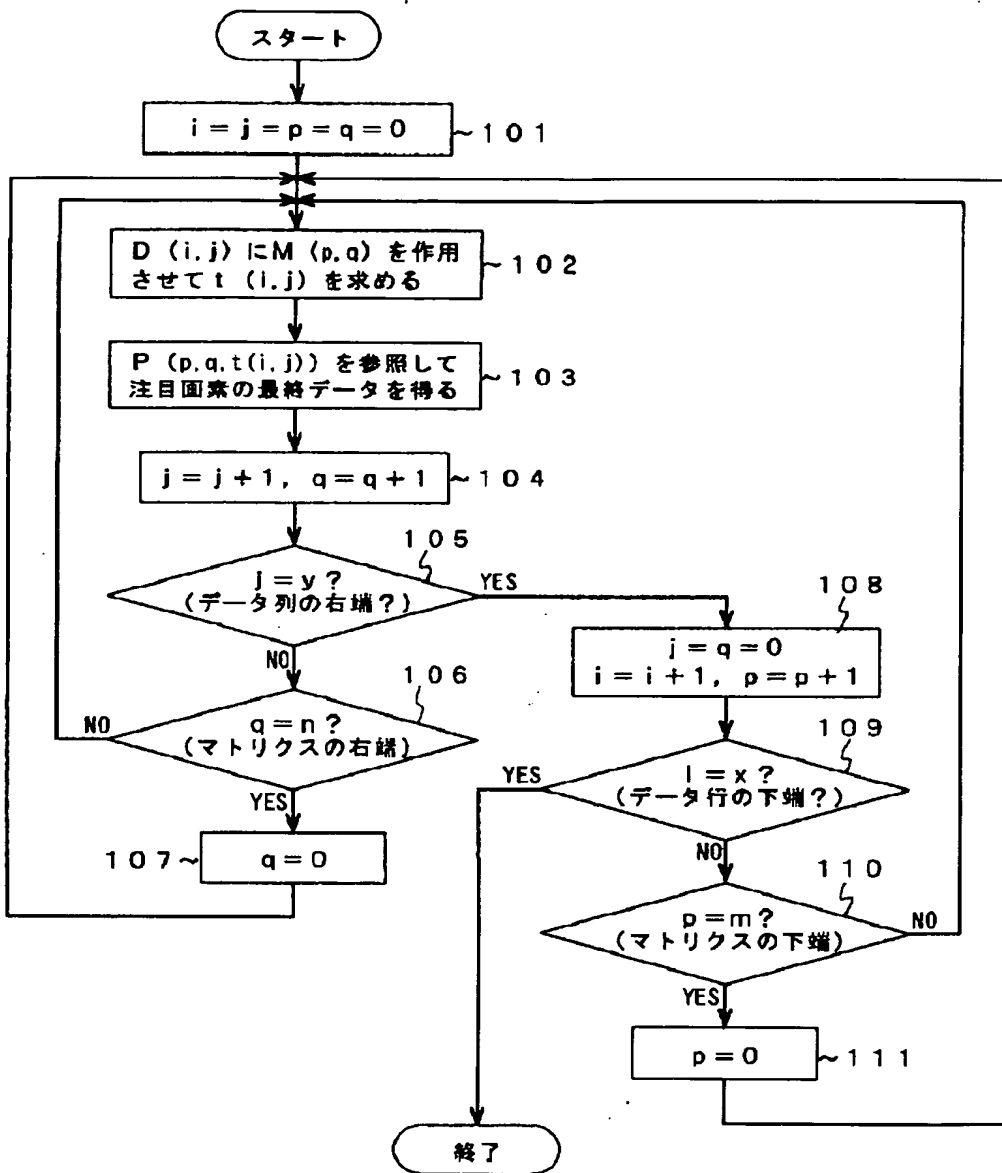
【図6】

符号パターンマトリクス

階調値	符号パターン	階調値	符号パターン								
0 →	<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0	0 →	<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0
0	0										
0	0										
0	0										
0	0										
1 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0	1 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0										
0	0										
1	0										
0	0										
2 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1	2 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1
1	0										
0	1										
1	0										
0	1										
3 →	<table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1	3 →	<table><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	1	0	1	1
1	1										
0	1										
1	0										
1	1										
4 →	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1	4 →	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1										
1	1										
1	1										
1	1										

階調値	符号パターン	階調値	符号パターン								
0 →	<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0	0 →	<table><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0
0	0										
0	0										
0	0										
0	0										
1 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0	1 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0										
0	0										
1	0										
0	0										
2 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1	2 →	<table><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1
1	0										
0	1										
1	0										
0	1										
3 →	<table><tr><td>1</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table>	1	0	1	1	3 →	<table><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1
1	0										
1	1										
1	1										
0	1										
4 →	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1	4 →	<table><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1										
1	1										
1	1										
1	1										

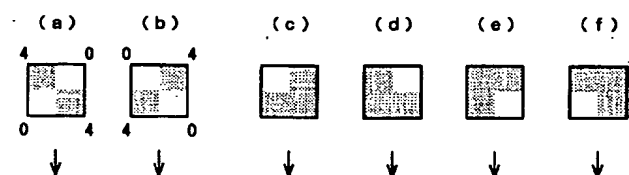
【図7】



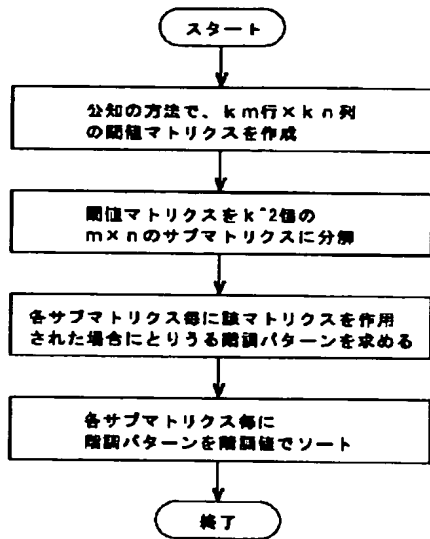
【図13】

1	9	3	11
13	5	15	7
4	12	2	10
16	8	14	6

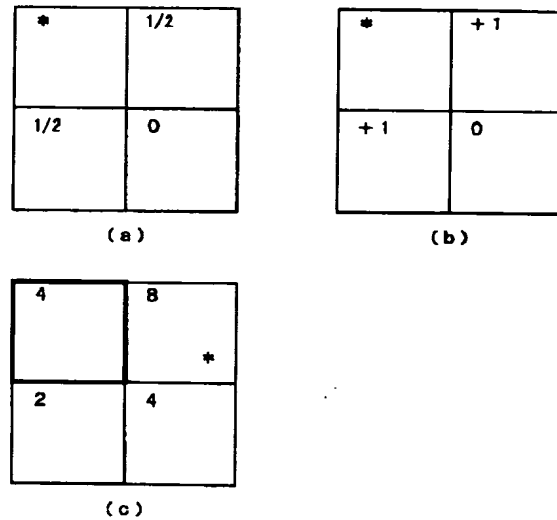
【図19】



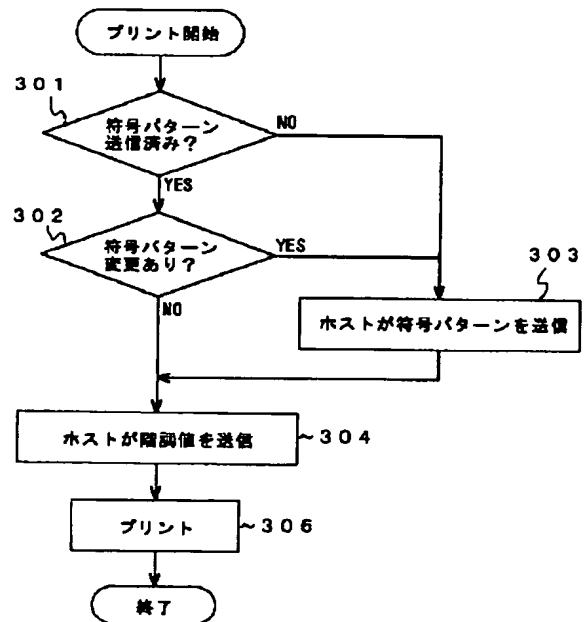
【図9】



【図11】



【図15】



【図14】

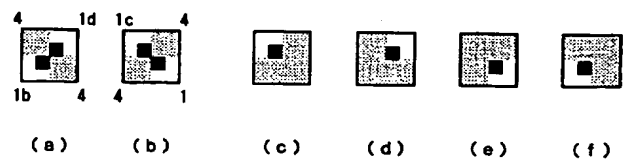
符号パターンマトリクス

随調値	符号 パターン				
0 →	<table border="1"><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0
0	0				
0	0				
1 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0				
0	0				
2 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1
1	0				
0	1				
3 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1
1	1				
0	1				
4 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1				
1	1				

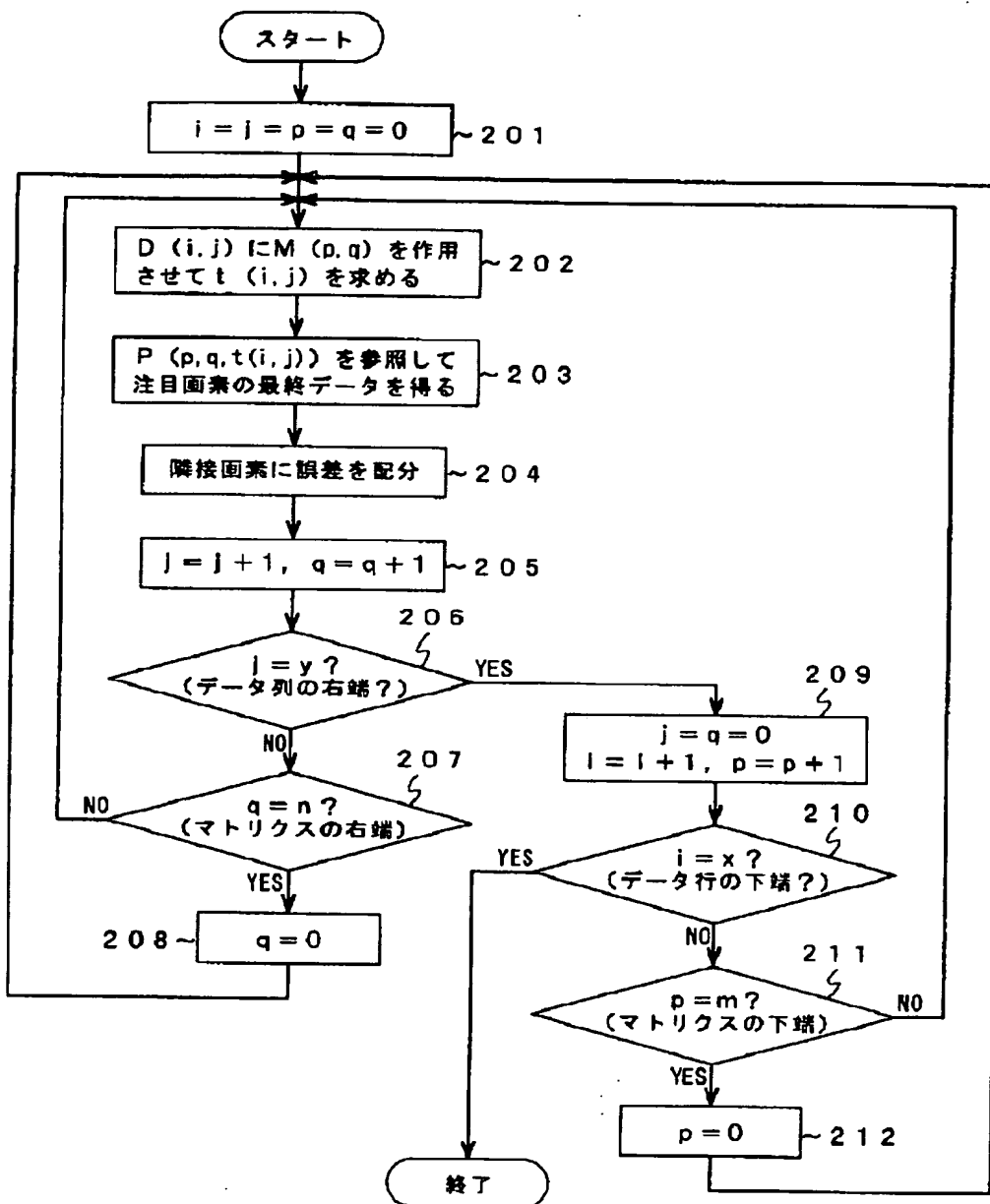
随調値	符号 パターン				
0 →	<table border="1"><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0
0	0				
0	0				
1 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0				
0	0				
2 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1
1	0				
0	1				
3 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1
1	1				
0	1				
4 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1				
1	1				

随調値	符号 パターン				
0 →	<table border="1"><tr><td>0</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	0	0	0	0
0	0				
0	0				
1 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>0</td></tr></table>	1	0	0	0
1	0				
0	0				
2 →	<table border="1"><tr><td>1</td><td>0</td></tr><tr><td>0</td><td>1</td></tr></table>	1	0	0	1
1	0				
0	1				
3 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>0</td><td>1</td></tr></table>	1	1	0	1
1	1				
0	1				
4 →	<table border="1"><tr><td>1</td><td>1</td></tr><tr><td>1</td><td>1</td></tr></table>	1	1	1	1
1	1				
1	1				

【図20】



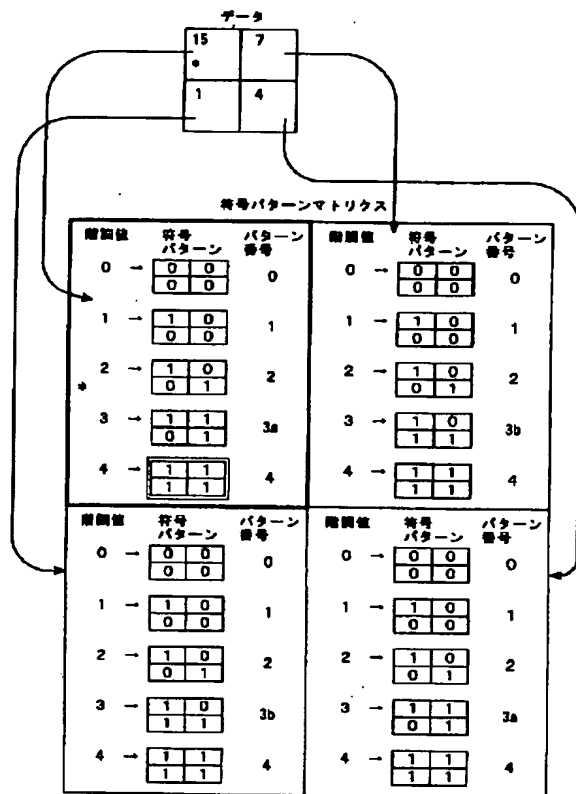
【図12】



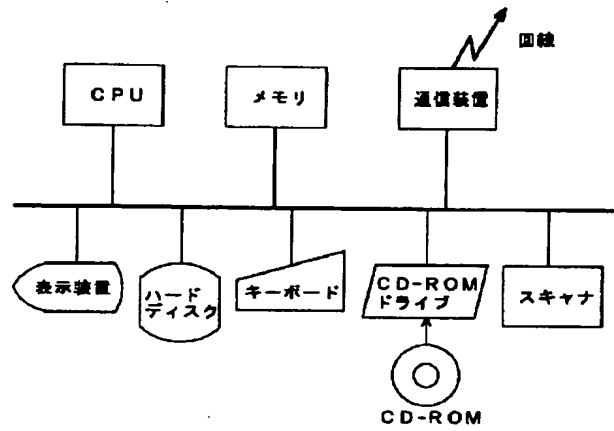
【図21】

追加符号 パターン	パターン 番号				
<table border="1"> <tr><td>0</td><td>1</td></tr> <tr><td>0</td><td>0</td></tr> </table>	0	1	0	0	1 b
0	1				
0	0				
<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td></tr> </table>	0	0	0	1	1 c
0	0				
0	1				
<table border="1"> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td></tr> </table>	0	0	1	0	1 d
0	0				
1	0				

【図16】



【図27】

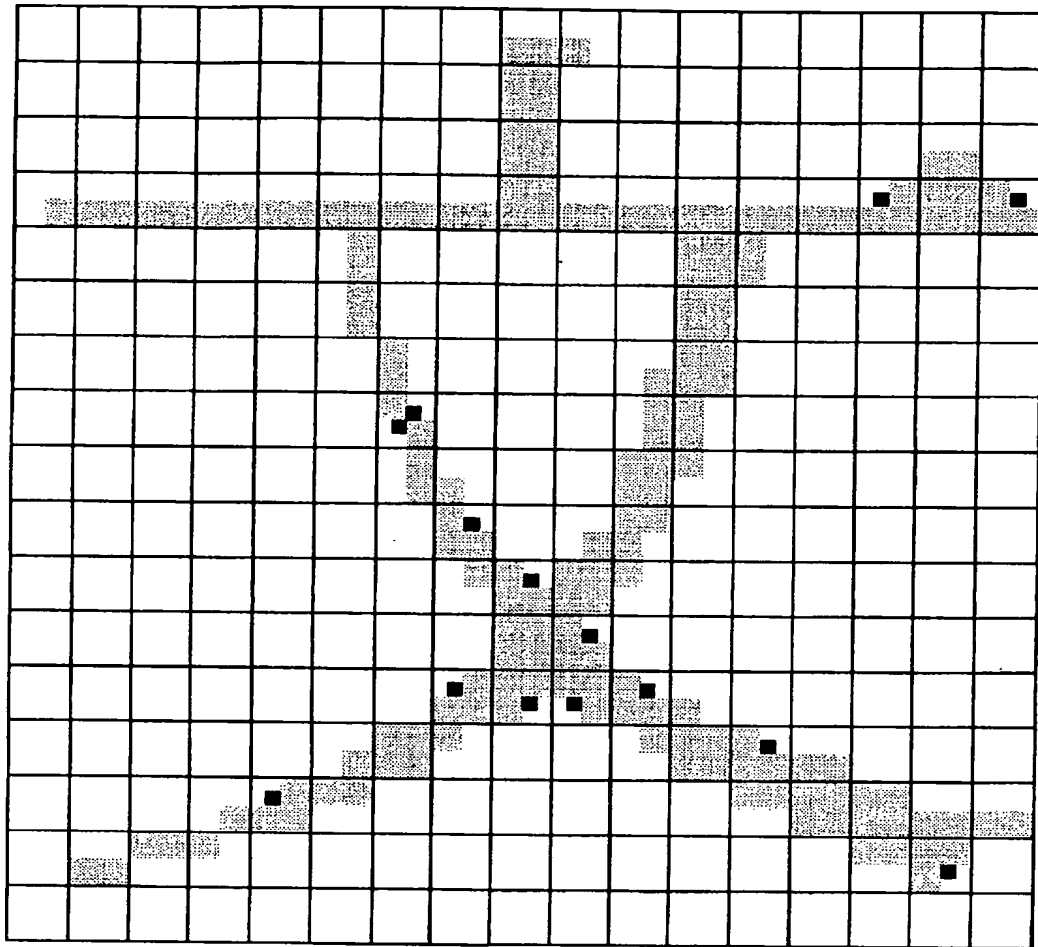






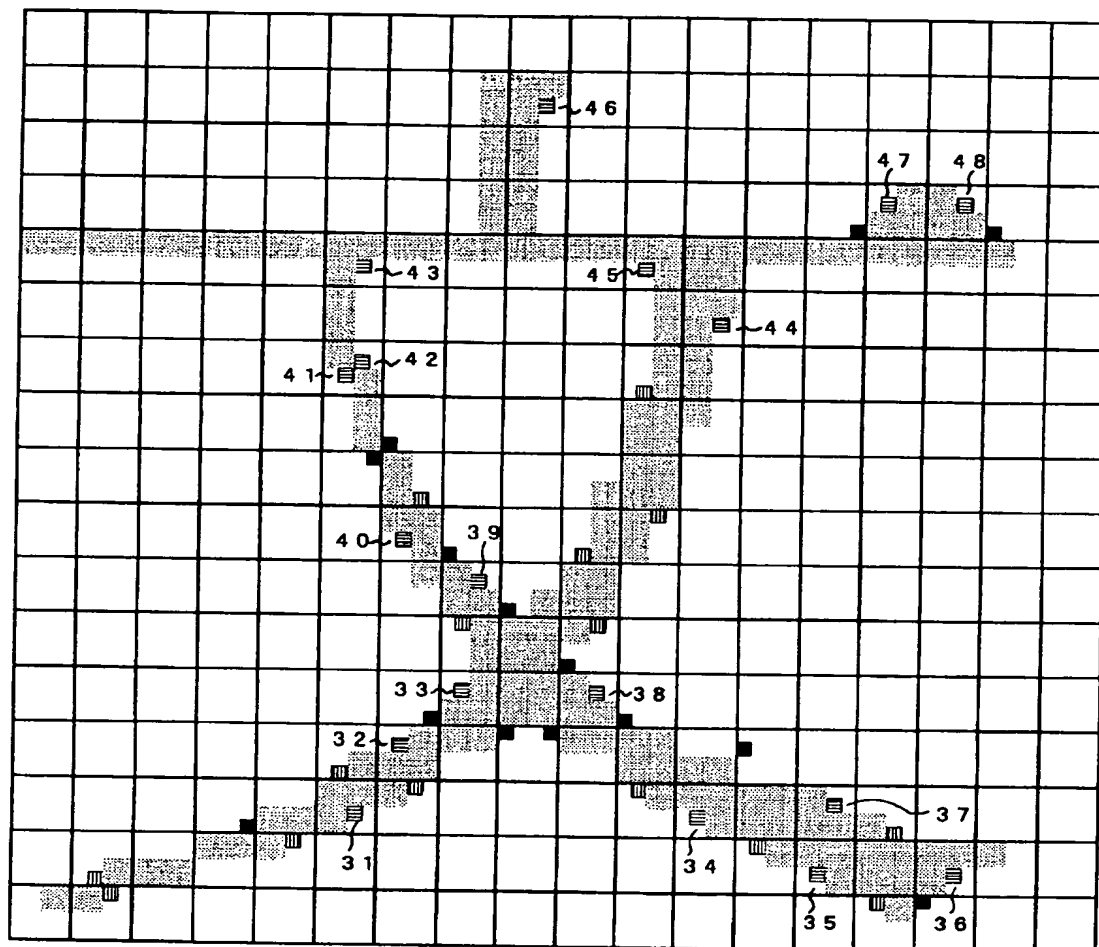
[illegible]

【図22】

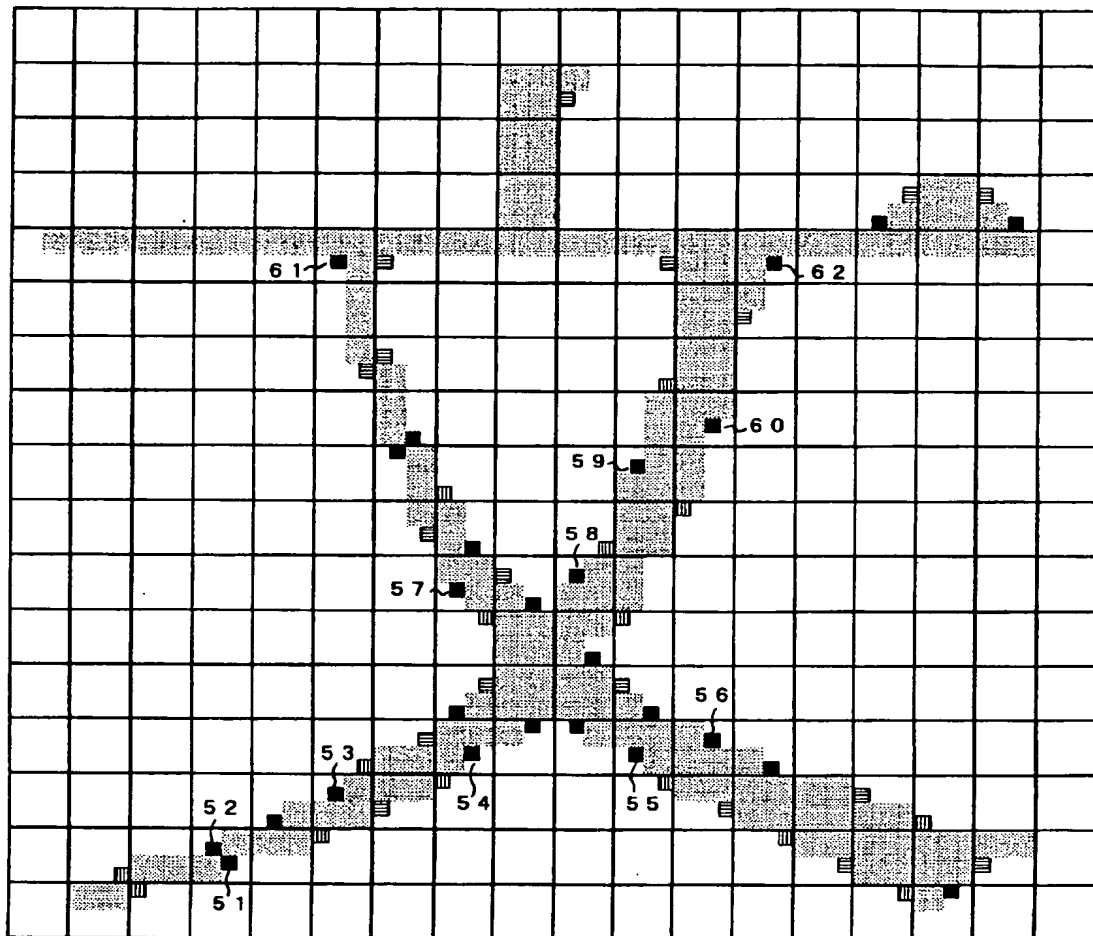




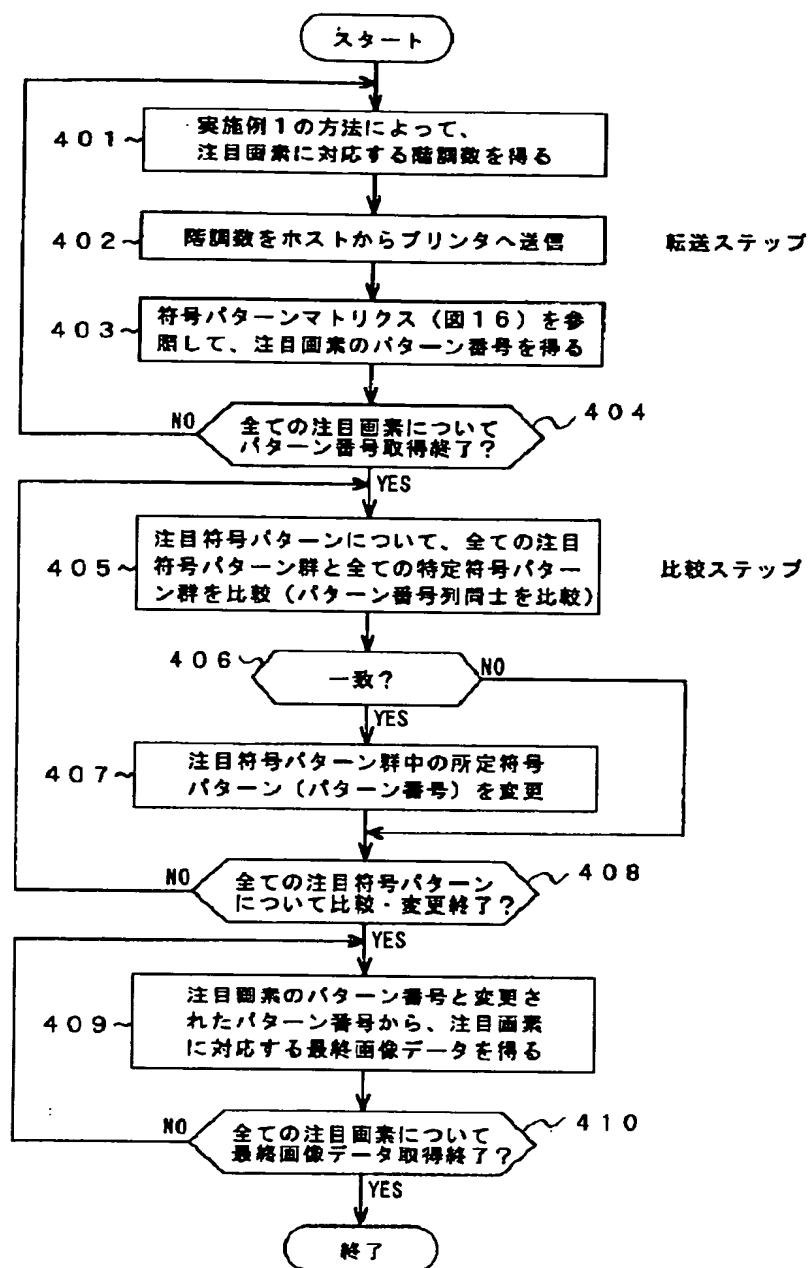
【図24】



【図25】



【図26】



## フロントページの続き

F ターム(参考) 5B057 BA29 BA30 CA01 CA02 CA08  
CA12 CA16 CB01 CB02 CB08  
CB12 CB16 CB18 CC02 CD05  
CE13 CE16 CH07 CH18 DA17  
5C076 AA21 AA27 BA06 BA07  
5C077 LL18 MP01 MP08 NN15 PP20  
PP55 PP68 PQ08 PQ20 RR04  
RR08 RR16

**JPO and INPIT are not responsible for any damages caused by the use of this translation.**

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. \*\*\*\* shows the word which can not be translated.
3. In the drawings, any words are not translated.

---

## CLAIMS

---

[Claim(s)]

[Claim 1] It is an image data generation method which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution, Two or more sorted thresholds corresponding to a position of this pixel are used for every noticed picture element of said 1st image data, An image data generation method generating said 2nd image data by changing a pixel value into a gradation number and referring to a predetermined code pattern corresponding to a position and said changed gradation number of said pixel.

[Claim 2] The image data generation method according to claim 1 applying an error diffusion process at the time of said gradation number conversion.

[Claim 3] The image data generation method according to claim 1 characterized by this arrangement serving as dot distributed type when said code pattern in the same gradation value has been arranged corresponding to a picture element position of said 1st image data.

[Claim 4] The image data generation method according to claim 1 characterized by this arrangement serving as a blue noise type when said code pattern in the same gradation value has been arranged corresponding to a picture element position of said 1st image data.

[Claim 5] The image data generation method according to claim 1 characterized by making said gradation number into a exponentiation of  $n$  when said gradation number conversion is performed by the host side, said gradation number is transmitted to an image output device and transmission of said gradation number is performed with  $n$  base.

[Claim 6] In an image data generation method which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution, Two or more sorted thresholds corresponding to a position of this pixel are used for every noticed picture element of said 1st image data, By changing a pixel value into a gradation number and referring to the 1st predetermined code pattern corresponding to a position and said changed gradation number of said pixel, An attention code pattern group of specified shape which is an image data generation method which generates said 2nd image data, and comprises a code pattern corresponding to said noticed picture element, and a code pattern corresponding to said 1st image data of a prescribed number, An image data generation method characterized by changing the 2nd predetermined code pattern of said attention code pattern group into the 3rd code pattern when a specific code pattern group is compared and it is in agreement.



[Claim 7]Perform said gradation number conversion by the host side, transmit said gradation number to an image output device, and this image output device generates said 2nd image data by referring to a code pattern corresponding to said gradation number, and. The image data generation method according to claim 6 making a change to said 3rd code pattern.

[Claim 8]The image data generation method according to claim 6, wherein said 3rd code pattern that is the target of said change differs from all of said 1st predetermined code pattern.

[Claim 9]The image data generation method according to claim 6 characterized by comparing an attention code pattern group and a specific code pattern group of this plurality using said attention code pattern group two or more.

[Claim 10]When said 1st image data has two or more color components and compares an attention code pattern group and a specific code pattern group of said specified shape for this every color component, The image data generation method according to claim 6 characterized by making the number of said specific code pattern groups less than the number of specific code pattern groups of other color components about at least one color component among said color components.

[Claim 11]When said 1st image data has two or more color components and compares an attention code pattern group and a specific code pattern group of this plurality for this every color component, using said attention code pattern group two or more, The image data generation method according to claim 9 characterized by making the number of said attention code pattern groups less than the number of attention code pattern groups of other color components about at least one color component among said color components.

[Claim 12]The image data generation method according to claim 10 or 11, wherein said at least one color component is yellow.

[Claim 13]It is the recording medium which recorded a program for making a computer realize a function which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution and in which computer reading is possible, A function to change a pixel value into a gradation number using two or more sorted thresholds corresponding to a position of this pixel for every noticed picture element of said 1st image data, A recording medium which recorded a program for making a computer realize a function which generates said 2nd image data by referring to a predetermined code pattern corresponding to a position and said changed gradation number of said pixel and in which computer reading is possible.

---

## DETAILED DESCRIPTION

---

[Detailed Description of the Invention]

[0001]

[Field of the Invention]This invention picture information about the recording medium which recorded the intermediate color processing, image data generation method, and

image data generation processing program which carry out definition conversion as data for output units, It is the art applied to device drivers, such as an application program, a display driver, and a printer driver.

[0002]

[Description of the Prior Art]By the spread of computers, the complicated document having contained vector data, such as a text and graphics, and image data, such as natural drawing, can create now easily. The resolution of a printer is also improving so that such a document can be printed by high definition, but [ as a result ] the data volume which should transmit to a printer increased and increase of processing time or transfer time is caused.

[0003]Typical methods in the case of outputting a document which was described above using the printer which has predetermined resolution and gray-scale-representation capability (the following, the 1st method) include some which are indicated, for example to JP,5-96792,A.

[0004]In this 1st method, within a printer driver, while carrying out raster deployment of the vector data as bit map data of prescribed resolution, After performing definition conversion to prescribed resolution to image data, such as natural drawing, gradation number conversion (intermediate color processing) to the number of specified floor tones is performed to both, and the data after gray scale conversion (after intermediate color processing) is transmitted to a printer. Therefore, in order to raise print speed, raster deployment, definition conversion, and intermediate color processing need to be accelerated, and it can come, simultaneously the data transfer to a printer also needs to be accelerated.

[0005]

[Problem(s) to be Solved by the Invention]Then, the method of accelerating raster deployment and intermediate color processing is indicated to "PPA Printer Software Driver Design" (HPJournal, June 1997).

[0006]In order to accelerate raster deployment and intermediate color processing in this method (the following, the 2nd method), Raster deployment and definition conversion of image data are performed in resolution lower than the resolution of a printer, After performing intermediate color processing, it transmits to a printer, and the firmware of a printer is used, and it is the whole definition conversion (in order to distinguish from the definition conversion of an image data simple substance). this --" of whole following" -- calling -- it carries out, and the above-mentioned low resolution is changed into the resolution of a printer, and is outputted (in addition, in the literature shown above, raster deployment of the text is carried out in one half of the resolution of a printer, and changing the resolution of a text twice using firmware is indicated).

[0007]As a result of reducing the data volume which should be processed if raster deployment, definition conversion of a bit map, and intermediate color processing are performed with a low resolution as described above, improvement in the speed is attained. However, when definition conversion is performed to the data after intermediate color

processing, after performing definition conversion, compared with the case where intermediate color processing is performed, image quality deteriorates easily clearly.

[0008]becoming the following tendencies if the 1st method and the 2nd method which were explained above are compared briefly -- an outline -- "-- the 1st method is image quality priority and the 2nd method can be said to be speed priority."

[0009]

[0010]This invention was made in consideration of the above-mentioned situation, and the purpose of this invention, Raster deployment and the definition conversion of an image are quicker than the 1st and 2nd conventional method, It is in providing the image data generation method and recording medium which serve as middle of the 1st and 2nd conventional method about intermediate color processing, the amount of data transmitting, and image quality and which have both strong point and perform intermediate color processing and the whole definition conversion simultaneously.

[0011]namely

[0012]

[Means for Solving the Problem]In order to attain said purpose, in the invention according to claim 1. It is an image data generation method which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution, Two or more sorted thresholds corresponding to a position of this pixel are used for every noticed picture element of said 1st image data, It is characterized by generating said 2nd image data by changing a pixel value into a gradation number and referring to a predetermined code pattern corresponding to a position and said changed gradation number of said pixel.

[0013]In the invention according to claim 2, it is characterized by applying an error diffusion process at the time of said gradation number conversion.

[0014]In the invention according to claim 3, when said code pattern in the same gradation value has been arranged corresponding to a picture element position of said 1st image data, it is characterized by this arrangement serving as dot distributed type.

[0015]In the invention according to claim 4, when said code pattern in the same gradation value has been arranged corresponding to a picture element position of said 1st image data, it is characterized by this arrangement serving as a blue noise type.

[0016]In the invention according to claim 5, when said gradation number conversion is performed by the host side, said gradation number is transmitted to an image output device and transmission of said gradation number is performed with  $n$  base, it is characterized by making said gradation number into a exponentiation of  $n$ .

[0017]In an image data generation method which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution in the invention according to claim 6, Two or more sorted thresholds corresponding to a position of this pixel are used for every noticed picture element of said 1st image data, By changing a pixel value into a gradation number and referring to the 1st predetermined code pattern corresponding to a position and said changed gradation number of said pixel, An attention code pattern group of specified shape which is an image data generation method which generates said 2nd image data, and comprises a code pattern corresponding to said noticed picture element, and a code pattern corresponding to said 1st image data of a prescribed number, When a specific code pattern group is compared and it is in agreement, it is characterized by changing the 2nd predetermined code pattern of said attention code pattern group into the 3rd code pattern.

[0018]In the invention according to claim 7, perform said gradation number conversion by

the host side, transmit said gradation number to an image output device, and this image output device generates said 2nd image data by referring to a code pattern corresponding to said gradation number, and. It is characterized by making a change to said 3rd code pattern.

[0019]In the invention according to claim 8, said 3rd code pattern that is the target of said change is characterized by differing from all of said 1st predetermined code pattern.

[0020]In the invention according to claim 9, it is characterized by comparing an attention code pattern group and a specific code pattern group of this plurality, using said attention code pattern group two or more.

[0021]When said 1st image data has two or more color components and compares an attention code pattern group and a specific code pattern group of said specified shape for this every color component by the invention according to claim 10, About at least one color component, it is characterized by making the number of said specific code pattern groups less than the number of specific code pattern groups of other color components among said color components.

[0022]When said 1st image data has two or more color components and compares an attention code pattern group and a specific code pattern group of this plurality for this every color component by the invention according to claim 11, using said attention code pattern group two or more, About at least one color component, it is characterized by making the number of said attention code pattern groups less than the number of attention code pattern groups of other color components among said color components.

[0023]In the invention according to claim 12, it is characterized by said at least one color component being yellow.

[0024]It is the recording medium which recorded a program for making a computer realize a function which generates image data of the 2nd resolution higher than the 1st resolution from image data of the 1st resolution in the invention according to claim 13 and in which computer reading is possible, A function to change a pixel value into a gradation number using two or more sorted thresholds corresponding to a position of this pixel for every noticed picture element of said 1st image data, By referring to a predetermined code pattern corresponding to a position and said changed gradation number of said pixel, it is characterized by being the recording medium which recorded a program for making a computer realize a function which generates said 2nd image data and in which computer reading is possible.

[0025]

[Embodiment of the Invention]Hereafter, one example of this invention is concretely described using a drawing.

(Example 1) Drawing 1 shows the composition of Example 1 of this invention. 1 in a figure Raster deployment or the image data after carrying out definition conversion, The threshold matrix by which the threshold was arranged 2, the code pattern storage for which the code pattern was stored 3, and 4 are printing data, the image data 1 and a threshold matrix are in the processing unit 5, and the code pattern storage 3 and the

printing data 4 are in the printer 6. In Example 1 of this invention, the printer of resolution 600dpi and a gradation number binary is used as an output unit, for example.

[0026]The 1st and 2nd conventional method hereafter mentioned above before explaining this invention is explained using figures. Drawing 2 shows the threshold matrix of 4x4 used by gradation number conversion (intermediate color processing). Each pixel of the data of a matrix takes the values (multiple value in which each pixel is 4 bits) from 0 to 15, for example.

[0027]Drawing 3 (a) is data after carrying out raster deployment of the vector data by 600dpi by the 1st above mentioned method, or data after carrying out definition conversion of the image data to 600dpi. Drawing 4 (a) shows raster deployment or data after carrying out definition conversion by 300dpi by the 2nd method similarly described above. In this example, drawing 3 (a) is the same thing as the case where the definition conversion of drawing 4 (a) is doubled by a publicly known these days side method.

[0028]Since the throughput of the raster deployment [ itself ] (or definition conversion) is proportional to the square of resolution in general, the throughput of raster deployment (or definition conversion) of the 1st method will be 4 times the throughput of the 2nd method.

[0029]In the 1st method, each pixel value is compared with the threshold of the threshold matrix of drawing 3 (b) corresponding to each picture element position, and the (same matrix as drawing 2) about the data of drawing 3 (a), and binary-ization is performed with a publicly known dither method. Since the value of the pixel which has a value beyond a threshold is outputted with 1 in the case of this example, drawing 3 (c) is obtained as a gradation value, but a total of 16 comparison operations is needed to a threshold. Since it becomes data (the same as that of drawing 3 (c)) which the data shown in drawing 3 (d) should print in not performing compression etc. to this day evening, to a printer, 2 bits [ 16 pixels x ] = 32 bits data will be transmitted.

[0030]Similarly, by the 2nd method, when making the threshold matrix of drawing 2 act to the data of drawing 4 (a), One fourth of the matrices of the matrix of 4x4 are used in order, and the matrix of drawing 4 (b) performs binary-ization in this example, The gradation value of drawing 4 (c) is acquired (to the data (a figure is omitted) of 2x2 which adjoins the right of the data of drawing 4 (a), the matrix (thresholds 3, 15, 11, and 7) of 2x2 which adjoins the right of the matrix of drawing 4 (b) in drawing 2 is used).

[0031]Since a total only of four comparison operations is performed to a threshold in the case of this 2nd method, there is little time which intermediate color processing takes compared with the 1st method. When not performing a data compression etc., 2 bits [ 4 pixels x ] = 8 bits data will be transmitted to a printer, and there are also few amounts of data transmitting.

[0032]However, in the 2nd method, it is the firmware by the side of a printer, and in order to double the resolution of the whole data of drawing 4 (c), the data which should be printed serves as drawing 4 (d). In this example, a publicly known these days side method is used as a definition conversion method.

[0033]Drawing 5 is a figure explaining this example, and drawing 5 (a) is the image data 1

of this example, and is data (2x2 pixels) after performing raster deployment or definition conversion by 300dpi like the 2nd method mentioned above. The threshold is arranged like drawing 5 (b) in the threshold matrix 2 of this example. And the image data 1 is changed into the gradation numbers from 0 to 4 by the threshold matrix 2.

[0034]Drawing 5 (b) is a matrix of 2x2 of four arrangement of an element, and the threshold is sorted by \*\*\*\* although four thresholds which constitute arrangement are the same as the threshold within each submatrix when the matrix of drawing 2 is decomposed into four submatrices of 2x2.

[0035]The gradation number conversion in this example compares the data of 1 pixel of drawing 5 (a) with the threshold of four pieces corresponding to the picture element position (comparing in order the threshold which is got blocked and by which drawing 5 (b) was sorted), and makes a gradation value (number-of-times of comparison-1) until a threshold exceeds picture element data. That is, for example, image data (15) is compared with the threshold (1, 5, 9, 13) of the threshold matrix to which it corresponds, and image data (7) is compared with the threshold (3, 7, 11, 15) of the threshold matrix corresponding to it.

[0036]As a result, drawing 5 (c) will be obtained as a gradation number corresponding to the data of drawing 5 (a), and that number of times of comparison will be a total of 11 times.

[0037]In this example, although the gradation value shown in drawing 5 (c) is transmitted to the printer 6, the data volume is 3 bits [ 4 pixels x ] (5 gradation) = 12 bits.

[0038]And in this example, five arrangement which is the code patterns of 2x2 is made into an element at the printer 6 side, and the code pattern matrix (drawing 6) of 2x2 corresponding to each position of four pixels of drawing 5 (a) is provided in the code pattern storage 3.

[0039]Each element of the code pattern matrix of drawing 6 sorts the pattern of 2x2 which can be taken in order of a gradation number, when the matrix of drawing 2 is decomposed into four submatrices of 2x2 and the data (1-pixel 16 gradation) of 2x2 is made to act on each submatrix. A code pattern is equivalent to the code book in vector quantization.

[0040]And to the noticed picture element (\* seal) of the value 15 of drawing 5 (a), five code patterns in the thick line in drawing 6 (\* seal) correspond, and the code pattern (1, 1, 1, 1) surrounded in firmware by the double line corresponding to the gradation value 4 of the noticed picture element is chosen. Drawing 5 (d) is obtained as the data 4 which the code pattern corresponding to the position and gradation value of the pixel is chosen, and should print other pixels eventually in a similar manner.

[0041]In this example, in order for reference of a code pattern to perform processing equivalent to the definition conversion for [ whole ] obtaining the final data of drawing 4 (d) by the 2nd method, the processing concerning "the whole definition conversion" is not more nearly high-speed than the 2nd method. However, final data drawing 5 (d) according to the method of this example if drawing 4 (d), 3 (d), and 5 (d) are compared, so that clearly, The 1st method may be matched in high definition (for example, the story tonality of original data is saved better) rather than the 2nd method (in this example, since it was

easy, drawing 3 and drawing 4 were made the relation which carried out definition conversion by the these days side method, but.). When both relation is not so, the last day evening of this example will be inferior to the 2nd method.

[0042]In the above example, when the comparison result of the throughput of three methods is summarized simply, it is as follows (the throughput of intermediate color processing is approximated by the number of times of comparison with a threshold).

[0043]

Therefore, it can be said that this example is a method having the strong point of both the 1st method and the 2nd method.

[0044]Drawing 7 is a processing flow chart of Example 1 of this invention. Carry out gradation number conversion by the threshold matrix of the  $m$  line  $xn$  sequence which makes an element arrangement of the threshold by which  $k^2$  (square of  $k$ ) individual was sorted in the image data of the  $x$  line  $xy$  sequence, and a gradation number is obtained (Step 102), With reference to the code pattern matrix of the  $m$  line  $xn$  sequence which makes an element arrangement of  $k^2+1$  code pattern of  $k \times k$ , the printing data of a  $k \times$  line  $xky$  sequence are obtained (Step 103). When the right end of a matrix and a lower end are arrived at, image data is similarly processed from (Steps 106 and 110) and the position



shifted  $m+1$  pixel  $n+1$  pixel, respectively, and the same processing is repeated until it reaches the lower end of image data (Step 109).

[0045]  $D(i, j)$  is data located in = (a line, a sequence)  $(i, j)$  in image data among a figure, and  $M(p, q)$ , Are a threshold array located in = (a line, a sequence)  $(p, q)$  in a threshold matrix, and  $t(i, i)$ , It is a gradation number obtained when  $M(p, q)$  is made to act on  $D(i, i)$ , and  $P(p, q, t(i, j))$  is one code pattern in a code pattern matrix which should be referred to to  $D(i, j)$ .

[0046] The threshold matrix of the  $m$  line  $xn$  sequence which makes arrangement of the sorted threshold an element, It can create by the process flow of drawing 8 from the threshold matrix of a  $km$  line  $xkn$  sequence, and the code pattern matrix of an  $m$  line  $xn$  sequence can be created by the process flow of drawing 9 from the threshold matrix of a  $km$  line  $xkn$  sequence.

[0047] In Example 1, in order to maintain the aspect ratio of original data, the size of the code pattern is made into  $k \times k$ , but when there is no necessity for maintenance, an aspect ratio should just create the code pattern which is not 1.

[0048] (Example 2) In Example 1, conversion to  $k^2+1$  gradation was performed using the arrangement of the sorted threshold of  $k^2$  pieces. Usually, when data transfer to a printer is constituted so that a gradation number may become a exponentiation of 2 here since it is performed with a binary system, (data transmitting/gradation number) serves as the minimum, and its transfer efficiency is good.

[0049] Since the gradation data for 4 pixels can be managed with 8 bits (= 4 pixels  $\times$  2 bits (4 gradation)) if drawing 10 is a threshold matrix which constitutes Example 2 and conversion to 4 gradation (0-3) is performed using the arrangement of the sorted threshold of three pieces, (Data transmitting/gradation number) can be made smaller than Example 1. Each submatrix of drawing 10 replaces the value of two centers of the threshold array of each submatrix of drawing 5 (b) by one average value of this 2 \*\* value.

[0050] (Example 3) Although the above-mentioned example performs intermediate color processing by a dither method at high speed than the 2nd method, it is applicable also to a publicly known error diffusion method. Drawing 11 is a figure explaining Example 3 at the time of applying this invention to an error diffusion method. Drawing 11 (a) is an error matrix of the common knowledge used in Example 3, \* Add the error ( $15-13=2$ ) produced in gradation number conversion of the noticed picture element (pixel value 15 of drawing 5 (a)) of a seal by halves to the right which adjoins a noticed picture element like drawing 11 (b), and a lower pixel, and perform gradation number conversion further to the noticed picture element (\* seal) of the data shown in drawing 11 (c). This example can also use the same code pattern matrix as drawing 6. Drawing 12 shows the process flow of Example 3, and Step 204 which distributes an error to processing of drawing 7 further is added.

[0051] (Example 4) Although Example 1 generated the code pattern matrix based on the matrix of drawing 2, If a code pattern matrix is generated based on the typical threshold matrix of dot distributed type as shown in drawing 13, the arrangement of the code pattern in the same gradation value itself will serve as dot distributed type like drawing 14. That

is, in drawing 14, if four code patterns corresponding to the gradation value 1 are made to correspond to the picture element position of original image data and are arranged, the matrix of 4x4 will be made, but this is dot distributed type. It has dot distributed type character [ the matrix of drawing 2 ].

[0052] Since all the arrangement of the code pattern which constitutes a code pattern matrix becomes the same so that clearly from drawing 14, in the case of the example of drawing 14, by holding only five kinds of code patterns by the printer side, While capacity of a code pattern matrix can be made small, it becomes high-speed [ reference of a matrix ].

[0053] (Example 5) It is also possible to generate a code pattern matrix by this invention again based on the publicly known blue noise type threshold matrix (random in the arrangement of a threshold) which is not illustrated. In this case, the arrangement of the code pattern in the same gradation value itself serves as a blue noise type. And in this example, the image quality which may rival error diffusion in the same processing speed as a dither method can be obtained.

[0054] About a blue noise type threshold matrix, "DigitalHalftoning Using. a Blue Noise Mask" and T.Mitsa. and KParker, 1991 International Conference on Acoustics, Speech, and Signal Processing, and February and May. 1991 (principle of the blue noise mask method, etc.) and "Modified Approach to the Construction of a Blue Noise Mask" by M.Yao and KParker, and Journal. Please refer to offElectronicImaging, Vol.3, No.1, 92-97, and January 1994 (method of creating a blue noise mask, etc.).

[0055] (Example 6) Although the printer 6 side held the code pattern matrix 3, it may constitute from an above-mentioned example so that the code pattern matrix 3 may be downloaded from the processing unit 5 (host) side. In changing the threshold matrix used for gradation number conversion, it accompanies by change of a code pattern. A host should just transmit a code pattern matrix to a printer in advance of transmission of gradation data in these cases.

[0056] Drawing 15 is a processing flow chart of Example 6. Before starting a print, a code pattern the host 5 Not ending with transmitting but the (step 301), When there is change to a code pattern (Step 302), a code pattern is transmitted to the printer 6 (Step 303), Then, a host transmits a gradation value to a printer (Step 304), and the printer 6 outputs printing data with reference to the code pattern corresponding to the transmitted gradation value (Step 305).

[0057] Thus, in this example, since the amount of data transmitting could be reduced since the gradation number was transmitted to image output devices, such as a printer, and the code pattern was made downloadable, it can respond to change of a code pattern.

[0058] (Example 7) and the 2nd method mentioned above in time can perform smoothing of a jaggy by performing template matching of a bit pattern as shown, for example in JP,2-112966,A. However, in performing template matching by bit patterns, a throughput increases and there is a problem of being easy to cause a speed fall.

[0059] So, in this example, the method of performing smoothing of a jaggy at high speed is proposed by performing template matching of the code pattern mentioned later, and

performing simple processing.

[0060]Hereafter, although this example is described, also in this example, the printer of resolution 600dpi and a gradation number binary is used as an output unit like Example 1. The threshold matrix to assume is the same as that of drawing 2, and uses the same code pattern matrix as drawing 14. However, in this example, as shown in drawing 16, what gave the pattern number (0, 1, 2, 3a, 3b, 4) to a different code pattern is used. The monochrome picture of a black chisel is assumed as original image data.

[0061]Now, drawing 17 and 18 carry out raster deployment of the vector font of the black character "sentence" by 300dpi, and the bit pattern (code pattern) acquired by the procedure of Example 1 is shown (in addition, although it is originally one figure). This is divided into two, drawing 17 is a left half of a "sentence", and drawing 18 is a right half of a "sentence." The value 0 in a code pattern supports the portion with the white value 1 in a code pattern, and in this example, since it is a processing result of a deep black character, as a gradation value, it is a gray portion with either 0 or 4.

[0062]The size of each pixel of 300dpi and 600dpi is shown in the upper left (a) of drawing 17, and the row of the code pattern of 2x2 corresponding to the pixel size of 600dpi and the row of the pattern number are shown in it at (b). In (b), the size of a code pattern in this example and the size of the code pattern group are shown.

[0063]In this example, the attention code pattern group of the square which comprises four code patterns containing an attention code pattern is compared with the specific code pattern group (drawing 19) of six kinds of squares, and when in agreement, an attention code pattern group is changed into the code pattern group shown in drawing 20, respectively. As for the shown portion, three of the code patterns (code pattern containing black) after change of drawing 20 do not exist in drawing 16 so that clearly from drawing 16, although it is black in drawing 20, and the changed position in a code pattern is shown. That is, although the code pattern of drawing 20 (e) exists as the pattern number 1 of drawing 16, there is no code pattern of drawing 20 (c), (d), and (f) into the pattern of drawing 16.

[0064]Therefore, it is used in this example, adding three code patterns (pattern numbers 1b, 1c, and 1d) shown in drawing 21. Since duplication arises in a code pattern so that clearly from drawing 16, the pattern number which eliminated duplication beforehand is defined as shown in drawing 16 and drawing 21. When the increase in the number of code patterns is not preferred, selection of not adding a code pattern is also possible.

[0065]Here, when the attention code pattern group of drawing 19 (a) is taken for an example, the relation between drawing 19 and drawing 20 is as follows. The specific code pattern group of drawing 19 (a) forms the permutation of a pattern number called 0, 4, 0, and 4 in the counterclockwise direction from the 1st quadrant. So, this is changed into the pattern number sequence (code pattern group) of 1d, 4, 1b, and 4 which are shown in drawing 20 (a) when an attention code pattern group forms the pattern number sequence of 0, 4, 0, and 4 in the counterclockwise direction from the 1st quadrant.

[0066]In this example, as how to take the attention code pattern group (window for

comparison) to one attention code pattern (\* seal of drawing 17 (c)), Four kinds (the thick line A, the double line B, the dotted line C, the triplet D), A, B, C, and D which are shown in drawing 17 (c), are adopted, and as how to take the attention code pattern itself, the lengthwise direction and the transverse direction are carried out to every other one as shown in \* seal of drawing 17 (d) (it is because every other one is enough). The lattice arranged at drawing 17 and the 18 whole is in agreement with how to take the attention code pattern group A to \* seal of drawing 17 (d).

[0067]In this example, the attention code pattern group (window for comparison) A is taken, to one attention code pattern, when in agreement with a specific code pattern group (drawing 19), an attention code pattern group is changed into drawing 20, and as compared with the attention code pattern group B, \*\*\*\*\* is repeated further.

[0068]Since change of the code pattern in the attention code pattern group A does not affect the comparison and change in the attention code pattern group B in this example, It takes with A, B, C, and D to one attention code pattern, and the disposal method of moving to the following attention code pattern after changing, comparison and also takes and compares A to all the attention code patterns, B is taken and compared to all the attention code patterns, and the result is the same also at flume \*\*\*\*\*.

[0069]Then, drawing 22 shows the processing result at the time of taking the attention code pattern group (window for comparison) A about all the attention code patterns, and shows the code pattern in which the black portion was changed. This shows that smoothing of a jaggy is made and there is an effect of a grade that the window for comparison is also only one case.

[0070]Similarly, the lattice of drawing 23 which showed the processing result at the time of taking the attention code pattern group (window for comparison) B about all the attention code patterns, and has been arranged at drawing 17 and the 18 whole corresponds with how to take the attention code pattern group B to \* seal of drawing 17 (d). 11 to 25 in a figure portions show the newly changed code pattern.

[0071]Similarly, drawing 24 and the processing result to which the attention code pattern group (window for comparison) D was taken are drawing 25 figures, and the processing result which took the attention code pattern group (window for comparison) C shows the code pattern in which the portion of 31 to 48 and the portions of 51-62 were newly changed, respectively. It is clear that the effect's of the part which uses two or more windows for comparison, and smoothing it is large.

[0072]Thus, according to this example, it turns out easy and that the remarkable smoothing effect is demonstrated by the simple method of comparison and detection of only six specific patterns (although the portion which evil produces by smoothing also exists, of course). It is small if it sees from the whole effect. And since it is a simple method, it becomes possible to perform smoothing at high speed.

[0073]What is necessary is to compare only the attention pattern group which can be taken in that case in the end of original image data, although four attention code pattern groups (window for comparison) may be unable to be taken altogether.

[0074]Drawing 26 is the processing flow chart of Example 7 explained above. First, the gradation number corresponding to a noticed picture element is obtained by the method explained in Example 1 (Step 401). The above-mentioned gradation number conversion is performed by the host side, and a gradation number is transmitted to a printer from a host (Step 402).

[0075]The following processings are performed by the printer. That is, the pattern number of a noticed picture element is obtained with reference to the code pattern matrix (drawing 16) corresponding to a gradation number (Step 403). The above-mentioned processing is performed about all the noticed picture elements (Step 404).

[0076]When all the attention code pattern group and all the specific code pattern groups (drawing 19) of A-D are compared (Step 405) and it is in agreement about an attention code pattern (Step 406), the predetermined code pattern (pattern number) in an attention code pattern group is changed (Step 407).

[0077]For example, by 0 (white), when an attention code pattern is in agreement with (f) of drawing 19, the pattern number of a noticed picture element changes the pattern number 0 of a noticed picture element into drawing 20 (f), that is, changes it into the pattern number 1b (drawing 21). For example, when the pattern number of an attention code pattern of a noticed picture element corresponds with (d) of drawing 19 by 4 (black), the pattern number 0 in an attention code pattern is changed into the pattern number of 1 d (drawing 21). The above-mentioned processing is performed about all the attention code patterns (Step 408).

[0078]The final image data of 600dpi corresponding to a noticed picture element is obtained from the pattern number (drawing 16) of a noticed picture element, and the changed pattern number (drawing 21) (Step 410).

[0079](Example 8) What is necessary is just to perform the above processing for every image component in the above-mentioned example, in the case of a color picture so that it may be publicly known although the monochrome picture of the black chisel was assumed. namely, data after performing raster deployment or definition conversion by 300dpi -- a publicly known method -- each color component (cyanogen), for example, C, M (magenta), Y (yellow), and K (black) -- each time -- what is necessary is to separate into a plain and just to process drawing 26 for every plain

[0080]here -- the above 4 -- although comparison using all (drawing 19) six specific code pattern groups may be performed about plain all, when searching for rapidity rather than the smoothing effect, about yellow with low visibility, the number of specific code pattern groups may be reduced. That is, the subset of a specific code pattern group is used. For example, about yellow, it stops only to comparison with two specific code pattern groups shown in drawing 19 (a) and (b). The process flow in this case should just replace the comparison step 405 in drawing 26 with "comparing the subset of all the attention code pattern groups and a specific code pattern group about an attention code pattern (pattern number sequences are compared)" only about yellow.

[0081]Or about yellow, it may stop to the comparison only using A, without performing

comparison using the four above-mentioned attention code pattern groups A, B, and C and D (drawing 17 (c)). It is because the window for comparison has the smoothing effect also by one case as mentioned above. The process flow in this case should just replace the comparison step 405 in drawing 26 with "comparing the attention code pattern group A with all the specific code pattern groups about an attention code pattern (pattern number sequences are compared)" only about yellow.

[0082]In the above-mentioned example, since the number of attention code pattern groups was lessened about yellow with low visibility, processing speed can be raised, suppressing the fall of the smoothing effect observed by the user. And if top priority is given to processing speed, about yellow, selection of not performing comparison with a specific code pattern group is also possible.

[0083]Although the above-mentioned example showed application to a character, this invention is applicable to not only this but a graphics image, natural drawing, etc.

[0084](Example 9) This invention is not limited to the above-mentioned example, but software can also realize it. When software realizes this invention, As shown in drawing 27, CPU, a memory, a display, a hard disk, The computer system which consists of a keyboard, a CD-ROM drive, a scanner, etc. is prepared, and the program etc. which realize the image data generation function of this invention are recorded on the recording medium which CD-ROM etc. can computer read. And when performing image data generation processing of this invention, the program recorded on the above-mentioned recording medium is read, and the processing capability of this invention is realized by including in a part of OS by making this into a printer driver. The original image data used as a processing object is stored in the hard disk, the processing above-mentioned to this is performed, and image data higher-resolution than original image data is outputted to a printer etc.

[0085]

[Effect of the Invention]As mentioned above, since according to the invention of claim 1 and 13 statements gradation number conversion is performed using the sorted threshold and a code pattern is referred to to the noticed picture element of original image data as explained, Image data higher-resolution than original image data can be generated at high speed, and a picture [ high definition / method / conventional / 2nd ] can be acquired.

[0086]Since an error diffusion process is applied at the time of gradation number conversion according to the invention according to claim 2, high definition image data is generable.

[0087]Since the dot arrangement of the code pattern in the same gradation value took arrangement of dot distributed type according to the invention according to claim 3, the number of code patterns can be lessened.

[0088]Since the dot arrangement of the code pattern in the same gradation value took blue noise type arrangement according to the invention according to claim 4, the high definition image data which is equal to a high speed and an error diffusion process is generable.

[0089]According to the invention according to claim 5, since the gradation number was made into the exponentiation of  $n$  when transmission of a gradation number was made

with n base, the amount of data transmitting per gradation number can be made into the minimum.

[0090]According to the invention according to claim 6, by template matching, since the 2nd predetermined code pattern of an attention code pattern group is changed into the 3rd code pattern, the image data which it is higher-resolution than original image data, and smoothing required is generable at high speed.

[0091]According to the invention according to claim 7, perform gradation number conversion by the host computer side, transmit a gradation number to an image output device, and an image output device generates image data higher-resolution than original image data with reference to a code pattern, and. Since a change to the 3rd code pattern is made, the final image data which could reduce the amount of data transmitting from a host to a printer, and smoothing required is generable at high speed.

[0092]According to the invention according to claim 8, since the 3rd code pattern that is the target of change differs from all of the code pattern corresponding to a pixel, as a result of extending a code pattern, the high smoothing effect can be acquired.

[0093]Since two or more attention code pattern groups are used according to the invention according to claim 9, the high smoothing effect can be acquired.

[0094]Since the number of specific code pattern groups was lessened about at least one of the color components of original image data according to the invention according to claim 10, processing speed can be raised suppressing the influence on the smoothing effect.

[0095]Processing speed can be raised suppressing the influence on the smoothing effect, since the number of attention code pattern groups was lessened about at least one of the color components of original image data according to the invention according to claim 11.

[0096]Processing speed can be raised suppressing the fall of the smoothing effect observed by the user, since one color component was made into yellow with low visibility according to the invention according to claim 12.

---

## DESCRIPTION OF DRAWINGS

---

[Brief Description of the Drawings]

[Drawing 1]The composition of Example 1 of this invention is shown.

[Drawing 2]The threshold matrix of 4x4 used by gradation number conversion (intermediate color processing) is shown.

[Drawing 3](a) - (d) is a figure explaining the 1st conventional method.

[Drawing 4](a) - (d) is a figure explaining the 2nd conventional method.

[Drawing 5](a) - (d) is a figure explaining this example.

[Drawing 6]A code pattern matrix is shown.

[Drawing 7]It is a processing flow chart of Example 1 of this invention.

[Drawing 8]It is a processing flow chart of threshold-matrix creation of Example 1 of this invention.

[Drawing 9]It is a processing flow chart of the code pattern matrix preparation of Example

1 of this invention.

[Drawing 10] It is a threshold matrix which constitutes Example 2 of this invention.

[Drawing 11] It is a figure explaining Example 3 at the time of applying this invention to an error diffusion method.

[Drawing 12] It is a processing flow chart of Example 3 of this invention.

[Drawing 13] The threshold matrix of dot distributed type is shown.

[Drawing 14] The code pattern matrix of Example 4 of this invention is shown.

[Drawing 15] It is a processing flow chart of Example 6 of this invention.

[Drawing 16] The code pattern matrix used in Example 7 of this invention is shown.

[Drawing 17] Raster deployment of the vector font of the black character "sentence" is carried out, and the left half of the code pattern obtained by the procedure of Example 1 is shown.

[Drawing 18] The right half of a "sentence" is shown.

[Drawing 19] (a) - (f) shows a specific code pattern group.

[Drawing 20] (a) - (f) shows the code pattern group after change.

[Drawing 21] The code pattern and pattern number which were added are shown.

[Drawing 22] The processing result at the time of taking the window A for comparison is shown about drawing 17 and all the attention code patterns of 18.

[Drawing 23] The processing result at the time of taking the window B for comparison is shown about drawing 17 and all the attention code patterns of 18.

[Drawing 24] The processing result at the time of taking the window C for comparison is shown about drawing 17 and all the attention code patterns of 18.

[Drawing 25] The processing result at the time of taking the window D for comparison is shown about drawing 17 and all the attention code patterns of 18.

[Drawing 26] It is a processing flow chart of Example 7 of this invention.

[Drawing 27] The composition of Example 9 of this invention is shown.

[Description of Notations]

1 Image data

2 Threshold matrix

3 Code pattern storage

4 Printing data

5 Processing unit

6 Printer

---

[Translation done.]